IOWA STATE UNIVERSITY
Digital Repository

Retrospective Theses and Dissertations

Iowa State University Capstones, Theses and Dissertations

1-1-2005

ISTOS (Iowa State Optical Simulator): design, architecture, and features

Yana Ong
Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/rtd

Recommended Citation

Ong, Yana, "ISTOS (Iowa State Optical Simulator): design, architecture, and features" (2005). *Retrospective Theses and Dissertations*. 19202.
https://lib.dr.iastate.edu/rtd/19202

This Thesis is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

ISTOS (Iowa STate Optical Simulator): Design, architecture, and features

by

Yana Ong

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Computer Engineering

Program of Study Committee:
Arun K. Somani, Major Professor
Ahmed E. Kamal
Ruan Lu

Iowa State University

Ames, Iowa

2005

Copyright © Yana Ong, 2005. All rights reserved.

Graduate College
Iowa State University

This is to certify that the master's thesis of

Yana Ong

has met the thesis requirements of Iowa State University

Signatures have been redacted for privacy

DEDICATION

To my mom and dad, who have supported me for more than two decades,
my family, who have believed in me, and Jemy, for being there for me.
This thesis is especially dedicated to God.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First, I would like to thank my major professor, Dr. Arun K. Somani, for believing in me and for allowing me to do what I was most comfortable with within my years of graduate study under his guidance. I absolutely agree with Rama and Srini; I am always amazed on your capability of capturing many totally different topics and switching between any of them within seconds. Although there were times that I might disagree with you, but you are definitely one of the most competent professors out of those I have known for more than a quarter century.

I would also like to thank my committee members, Dr. Ahmed E. Kamal and Dr. Ruan Lu, for their willingness to share their time and their undoubtful consent to be in my committee.

I am also thankful to all the professors I have became acquainted with during my graduate study, either from my courses, seminars, or other encounters. They have all, one way or another, contributed to my achievement in being a more knowledgeable person.

Many thanks also go to my colleagues and friends, Jing, Sam, Srivatsan, and Varun, who have supported me during my ups and downs (especially downs) all these years. I loved the time when we still shared our office in the lab. There were stories, laughters, arguments, and endless memory that I would always remember. Thanks also to Mike Frederick, Nathan, Wensheng, Pallab, Rohit, Jinran, Amy, Ganesh, Mike Bezdek, and David, for the fun moments we shared on our weekly seminars. I would like to personally thanks David for his help and inputs to the development of ISTOS, and his willingness to overtake the project ongoing maintainance process.

Endless thanks to my friends who have always been there with me all these years and stood besides me when I need them. Joyi and Carrie, thank you for the funs and supports you have shown me. I definitely appreciate your willingness to share your precious time with me and bear my hardly-ending naggings and comments. I would also like to thank the other family of colleagues I have become close to: Mike Reid, Jinson, Raegan, Rius, Luis, and Ben. It was really fun to share stories and spend time with all of you within the short period of our acquaintanceship.

Thank you also goes to my wonderful friends, Venina & hubby, Lia & hubby, and Irene. Although you guys are far in East and West Coast, you have always been there for me and I know you will always be. Your wishes and prayers will always be remembered and kept in my heart.

Last but not least, I would like to thank the people I love most in this world: my mom, dad, my sisters and brother, and Jemy. Thank you for all the love and support you have given me, and for acknowledging my abilities without a doubt. My achievements are my way of saying thanks to you.

ABSTRACT

Wavelength division multiplexed (WDM) networks have emerged as the viable solution to meet the increasing demands of bandwidth due to tremendous growth in Internet and World Wide Web (WWW) usage. WDM networks divide an optical fiber bandwidth into multiple WDM channels called *wavelengths*. Current fiber technology allows a transmission capacity of up to 40 Gbps on a single wavelength while end-user requirement ranges from 45 to 622 Mbps typically. *WDM grooming networks* support this sub-wavelength bandwidth requirement by allowing multiple connections to share single wavelength. The minimum connection granularity on WDM grooming networks is called a *channel*.

Many different routing and channel assignment schemes and network survivability mechanisms in WDM grooming networks have been proposed in research literature. The routing and channel assignment algorithms select the path and the channels on each link of the path a connection request is routed. The network survivability algorithm identifies the strategy used to protect the network against single link or single SRLG failure. A common platform to compare these schemes, against the same set of traffic, is desirable.

In this thesis, we present a software tool that simulates routing and channel assignment as well as network survivability algorithms in WDM grooming networks, and serves as a framework for comparing multiple algorithms run under one simulation environment. The Iowa STate Optical Simulator (ISTOS) tool allows simultaneous simulation of multiple networks with the same topology but different routing and channel assignment and/or network protection and recovery schemes. The networks are simulated with common simulation parameters that determine the network traffic rate, the failure rate (if applicable), the total number of connection requests to be injected into the simulation, and the maximum bandwidth that can be requested by each connection (referred as *request granularity*). This

allows parallel comparison of the schemes which could assist in identifying the most efficient traffic provisioning algorithm that best suits a particular network topology under certain traffic (and failure) pattern.

CHAPTER 1 Introduction

There has been a significant increase in bandwidth demand during the last two decades. The main driver to this is the tremendous growth in the usage of Internet and World Wide Web. Technology advances also result in lower cost of bandwidth and hence encourage the development of applications needing higher bandwidth, such as video and audio streaming.

Optical fiber offers much higher bandwidth than copper cables and has low attenuation and bit error rate. It is able to transmit a light signal for a relatively long distance without having to regenerate the signal. This makes optical fibers a natural choice for medium to transmit bit information in high-speed communication networks. Because of this and the improvement of optical components technology, optical networks are widely deployed as back-bone communication networks today.

1.1 WDM Networks

To improve the transmission efficiency over a single optic fiber, wavelength division multiplexing (WDM) technique is adopted. In WDM scheme, lower-rate data streams are multiplexed into a higher-speed transmission bit rate by carrying the data simultaneously over the multiple wavelengths. Optical WDM network supports the lower-rate end-user bandwidth requirement by dividing the higher-rate optical fiber bandwidth into multiple channels carried by multiple wavelengths.

A network consists of nodes interconnected by links. A node in WDM network usually represents an optical switch. A WDM network may employ a single or multiple fibers on a link between two nodes. The latter is called *multi-fiber WDM networks*.

In WDM networks, data is transmitted on a *lightpath*, which is an optical end-to-end connection over a wavelength that may pass through several intermediate nodes. These networks are also known as *wavelength-routed networks* for obvious reason. Data streams over multiple wavelengths are multiplexed at source node and are demultiplexed at the destination node before retrieving the data. Lightpaths are routed and switched from one link to another link at intermediate nodes. In wavelength-routed networks, different lightpaths must be carried over different wavelengths if they share common links. If wavelength conversion is not available in the network, a lightpath has to utilize the same wavelength on all links along its route. This is known as the *wavelength continuity constraint.*

Wavelength continuity constraint still applies in multi-fiber networks if no wavelength conversion capability is provided within the networks, but a wavelength on a link from an input fiber can be switched to any of the fibers on the output link as long as it remains on the same wavelength.

1.2 WDM Grooming Networks

Present day fiber technology allows a transmission capacity of up to 40 Gbps on a wavelength. End-user requirements, on the other hand, typically range from 45 to 622 Mbps. These sub-wavelength bandwidth requirements call for an alternative traffic provisioning method. *Traffic grooming* allows the merging of smaller capacity requirements into higher capacity lightpaths and thus allows the optimal use of wavelength capacity.

Traffic grooming is achieved by dividing a wavelength into multiple time slots which are then multiplexed onto the wavelength. This technique is being employed in *WDM-TDM networks* or WDM grooming networks.

WDM grooming networks can be classified into two categories: dedicated-wavelength grooming (DWG) networks and shared-wavelength grooming (SWG) networks. In DWG, each source-destination node pair is connected by lightpath(s). A new lightpath is established

at the source if the connection request requirement cannot be satisfied by any of the existing lightpaths to the destination. In SWG, if none of the existing lightpaths at source to the destination can accommodate the request, it is multiplexed onto an existing lightpath to an intermediate node and then switched and routed to the destination either directly or through other intermediate nodes.

Grooming in SWG can be done in a static or dynamic manner. In static grooming, the source-destination pairs whose connection traffic are to be groomed are pre-determined, while these pairs are determined depending on the network state in terms of provisioned lightpaths upon the arrival of connection requests in dynamic SWG grooming.

1.3 Network Survivability

Failures are quite common in optical networks. Links may fail due to fiber cuts. Network failures may occur due to equipment failures including transmitters, receivers, and controllers. Node failures may also be caused by catastrophic incident such as fires or flooding in central offices. A group of links may share resources in the network or are laid on a common pipe or conduit, and this would result in a failure of more than one link at an instant. These failures are modeled as *Shared Risk Link Group (SRLG)* [13].

As the amount of data transmitted over a network increases, the issue of service disruption due to the network failures becomes more important. It is therefore significantly important to design resilient and reliable optical networks as these networks carry massive amounts of data.

Network survivability incorporates protection mechanisms to ensure fault tolerance in optical networks. These protection schemes usually involve providing redundant capacity within the network and in case of failures, restore the compromised connections by rerouting them using this redundant capacity [16].

Network protection schemes can be classified into two categories as described next:

1.3.1. Link Protection

Link protection schemes route a connection around a failed link. In case of a failure, the node connected to the failed link re-routes the connection around the failed link to the neighboring node on the original path. This scheme thus appears to be transparent to the source node, except in the case where the link connected to the source node fails.

Link protection may be achieved in either fiber or connection level. The first is called the fiber switched link protection, while the latter is referred as connection switched link protection. In the *fiber switched link protection (FSLP)* scheme, each link has a primary and a spare fiber [9]. On any link failure, all connections established on the primary fiber of the failed link are switched to the spare fiber present in the links along the fixed backup path.

To increase the resource efficiency, the primary and spare fibers may both be used for normal network operation to establish primary connection in the *connection switched link protection (CSLP)* scheme [9]. In this scheme, a consistency in the channel assignment between the links of the backup path and any other links on the original path has to be maintained.

Figure 1.1 shows a network with a primary connection established on path 1-2-3-6. When link 2-3 fails, connection on failed link is re-routed along links 2-4-5-3. It is always established on spare fibers of links 2-4, 4-5, and 5-3 in FSLP scheme. If CSLP scheme is utilized, connection must be routed such that channels used on the remaining working links on original path are maintained.

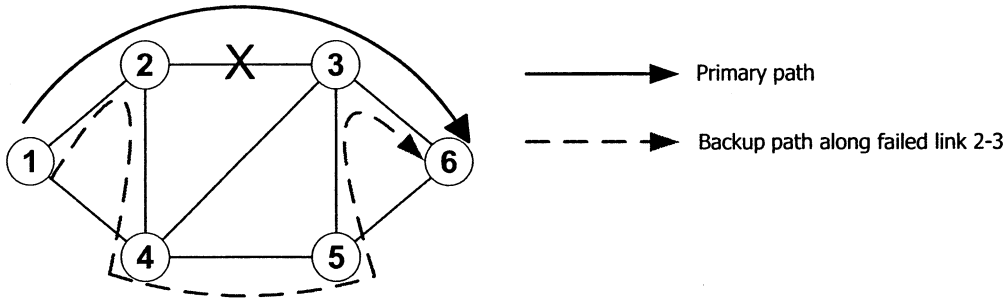


Figure 1.1. An example network showing a connection from 1 to 6 is established on a primary path of 1-2-3-6 and a backup path 1-2-4-5-3-6 routed along a failed link 2-3.

1.3.2. Path protection

In general, path protection schemes try to re-establish a failed connection on a provided backup path from source to destination node that may be independent of the primary path. According to their knowledge on the link failure, path protection schemes may be classified into two categories: failure-independent path protection and failure-dependent path protection.

In the *failure-independent path protection (FIPP)* [23], a backup path that is link-disjoint with the primary path is assigned. The backup path can then be used for any link failure, and hence no precise knowledge of the link failure is necessary. In the *failure-dependent path protection (FDPP)*, a backup path is assigned to each possible failure scenario [23]. The failure scenario considered is dependent on the primary path assigned to the connection, and therefore, precise knowledge of network failure is required.

Figure 1.2 shows the backup path established on network shown in Figure 1.1 using FIPP strategy. Note that the links on this backup path are disjoint from those on the primary path of 1-2-3-6. Thus, this backup path can be used in any link failure scenario, while the backup path shown in Figure 1.3 can only be used for either link 1-2 or link 2-3 failure scenario.

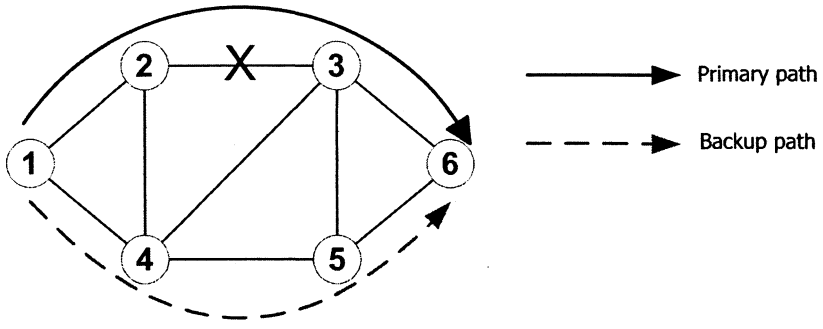


Figure 1.2. Backup path using FIPP strategy.

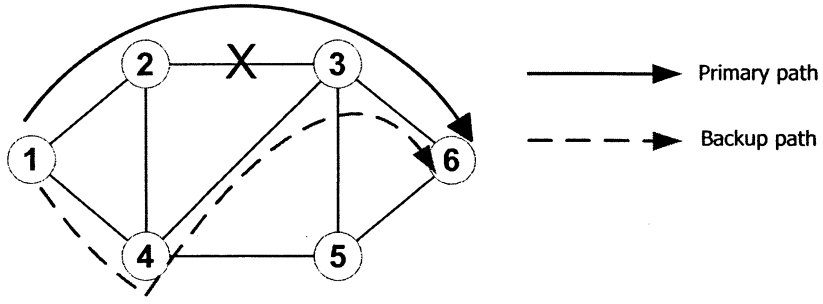


Figure 1.3. Backup path using FDPP strategy.

1.4 Issues and Motivation

A network needs to be designed carefully to meet budget or minimize network cost, to maximize resource utilization, to be resilient to failures, and for efficient operation and maintenance process. The optical WDM network design problem involves solving the lightpath topology design (LTD) problem and finding the routing and wavelength assignment (RWA) solution.

The LTD problem deals with finding an optimal lightpath topology given certain network cost and the higher-layer traffic requirements constraints. A *lightpath topology* or *virtual topology* is a graph consisting of network nodes and edges connecting two nodes in which each represents the lightpath between those two nodes [16]. Given the physical network topology and a set of end-to-end ligthpath connection requests (which could be obtained by

solving LTD problem), the connections' paths and their corresponding channels are determined after solving the RWA problem.

The RWA problem in wavelength-routed networks has been the focus of many researches, [1]-[5], [7], [10], [15], and [24] being a few. When connection requests traffic is known in advance, the RWA problem can be solved through integer linear programming (ILP) optimization formulation. RWA heuristics have been developed over the years to avoid the complexity of solving the ILP formula and to deal with unknown traffic requirements [4], [5], [10], [15], [24]. Given a dynamic set of request traffic, i.e. connection request arrives one by one at random, paths and channels have to be assigned while minimizing the blocking probability of future connection requests.

Network operation is another important issue in optical networks. Besides traffic handling and network monitoring for proper functionality, network reconfiguration in case of failures is also an important problem a network operator has to address. How fast a network is reconfigured when failures occur deduces how well the network is operating since it determines the service availability and length of service interruption period [16].

1.5 Contribution of the Thesis

Simulation provides a cheap and easy way of evaluating a proposed protocol and hence plays an important role in network design. In this thesis, we describe an advanced network evaluation tool, called Iowa STate Optical Network Simulator (ISTOS), which has been developed to assist network providers in addressing the network design and operational issues. It helps in identifying resource budgeting and evaluating technological choices of different routing and channel assignment schemes as well as fault-tolerance strategies in WDM grooming networks in order to design resilient and cost-effective optical networks.

One way of evaluating how well a new protocol performs is to compare it with other established protocols. A common simulation framework is hence desirable for comparing and

analyzing the performance of multiple protocols. ISTOS provides modeling of a WDM grooming network consists of nodes with various grooming capabilities and serves as a common platform for performance comparison of multiple networks with different design and operational schemes.

This thesis is organized as follows: Chapter 2 presents the complete list of network models, framework, and algorithms that ISTOS employs and provides. Chapter 3 describes the key contributions that ISTOS tool provides. Chapter 4 defines the performance metrics that ISTOS presents at the end of simulation and that are useful for evaluation of algorithms and schematics. Chapter 5 presents the architectural design of ISTOS. Chapter 6 presents a short instruction on how to use ISTOS through an example and briefly discusses on performance results on three different networks: NSFNet, 3x3 mesh torus, and a general topology. Chapter 7 presents our conclusions and possible future work.

1.6 Summary

In this chapter, we presented some background and issues that motivated the deployment of WDM grooming networks. We identified researches that have been done to address the design and operational issues on these networks. We identified the need for a common platform to compare these strategies, and the development of a simulation tool that provides such platform is the contribution of this thesis.

CHAPTER 2 Modeling in ISTOS

ISTOS provides a platform to model and design the network layer in WDM grooming networks. Given a network with its nodes and edges, in which each indicates the presence of optical fiber(s) connecting two nodes, and a (projected) traffic rate, ISTOS models the establishment of a connection by selecting a path and assigning one or more channels on every link on the chosen path upon the arrival of the connection request. Hence, it allows performance evaluation of the routing and channel assignment schemes as well as network survivability techniques by simulating random and dynamic connection request traffic into the networks under investigation.

2.1 Network Topologies

ISTOS models the node and link structures that form WDM grooming networks. A node in ISTOS represents an optical equipment in the network layer of optical system considered. This may be an optical switch or optical cross-connect that exists in a backbone network of smaller client networks such as SONET, ATM or IP. A link or edge between two nodes exists if there is one or more optical fiber that connects the two nodes.

2.1.1 Nodes in ISTOS

The level of grooming that a node is able to handle is determined by the type of switching equipments that are available at that node. The types of switching that can be modeled at a

node in ISTOS are: fiber, band, wavelength, and time-slot. A link between any two nodes may contain more than one *fiber*, each of which consists of multiple *wavelengths*. A set of wavelengths can be bundled together as a *band*. A band is a collection of wavelengths grouped together to improve network performance and management. A typical example of grouping wavelengths into bands is to reduce the number of multiplexers/demultiplexers needed in an optical network system with a large number of wavelengths [16]. Each wavelength can be divided into frames with multiple *time slots* in each frame. A *channel* is defined as a specific time slot on successive frames.

A fiber switch allows signal transmitted on a specific channel on a fiber on the input link to be switched to any of the fibers on the output link. A band, wavelength, and time-slot switch/interchanger allows similar switching capabilities on band, wavelength, and time-slot level, respectively. According to wavelength conversion capability at a node, there are two categories of wavelength converters: full-wavelength converter and limited-wavelength converter. *Full-wavelength converter* can switch signal on a wavelength on the input link to any wavelength on the output port, while *limited-wavelength converter* can only switch it to a limited set of wavelengths. ISTOS assumes any full wavelength conversion capability available at a node.

The switching/interchange capability at a node in turn determines its grooming capability level. If the wavelength converter is not available at a node, the node can switch traffic from an input link to an output link only if it is on the same wavelength; this node is referred to as *wavelength-level grooming node*. If no time-slot interchanger is available, a node can switch connections only if it obeys time-slot continuity constraint, i.e. time-slot of input link has to be the same as the time-slot of output link, and the node can only groom in *time-slot level*.

In general, each node in ISTOS is said to groom at trunk level. A *trunk* is defined to be a set of channels that can be interchanged at a node. Thus, a switching node in ISTOS is composed of several trunk switches. Signals cannot be switched across trunks, but connection on a channel in a trunk can be switched to any other available channels on the

same trunk. This means a connection has to obey *trunk continuity constraint* and trunk switches contained in a node are assumed to be full-permutation switch.

ISTOS allows modeling of dynamic shared-trunk grooming networks that does not constrain channel multiplexing onto a lightpath to be available only on connections with the same source and destination nodes. Shared-trunk grooming network is a generalized form of the shared-wavelength grooming (SWG) network.

2.1.2 Links in ISTOS

Two nodes in ISTOS network can be connected by at most one link. This link may be a unidirectional, bidirectional, or two-unidirectional link. Both bidirectional and two-unidirectional links are modeled as two links in opposite directions in the logical representation of the network topology. A bidirectional link shares its resources, i.e. bandwidth, among its two opposite-directional virtual links, while the amount of resources specified in ISTOS for a two-unidirectional link is specific for link in each direction.

The number of channels that form a trunk at a link and the number of trunks at a node are determined by two factors:

1. Properties of the link

   These properties include: the number of fibers in a link, number of bands per fiber, number of wavelengths per band, and the number of time-slots per wavelength.

2. Grooming capability of the nodes connected to the link

   Together with the link properties, the grooming level of the nodes determine whether the signal transmitted on a channel on the input link can be switched to another channel on different fiber, band, wavelength or time-slot on the output link.

2.2 Virtual Topologies

In ISTOS, a network structure is specified using an intuitive graphical user interface (GUI). The structure is converted into its corresponding virtual or logical topology by the software. Each node is identified by its grooming or switching capability, its number of trunks, and each trunk capacity. Each link is identified by the number of channels that are free on each subtrunk. A *subtrunk* is a set of channels that fall under the definition of a pair of trunks at nodes connected by the link.

The total capacity of the maximum number of channels on all subtrunks in a link is the product of the number of fibers on the link, number of bands per fiber, number of wavelengths per band, and the number of time-slots per wavelength. The grooming capability of its two end nodes determines the number of subtrunks and the capacity of each subtrunk.

Traffic in terms of connection requests is generated dynamically and injected into the network to analyze the performance of the network employing a particular routing and channel assignment schemes. In dynamic traffic, one connection request is generated at a time and each established connection is released after some finite time. Traffic is generated using the arrival rate following Poisson distribution with exponentially distributed holding time of mean 1.0. A request can go from any node to any other node with equal probability, and the possible source and destination nodes are selected depending on the node type – either a source, destination, or source and destination node – specified by users as part of node's parameters.

Failures can also be injected into the network to analyze how well a network withstands failures and how much performance degradation is faced by the network upon failure occurrences.

ISTOS supports the truncation routing. If *truncation routing* is employed, any trunk capacity of a link on selected path that is larger than the capacity requested by connection

request is truncated to the requested capacity. The effect of truncation on the performance of the network can therefore be analyzed.

ISTOS supports the visualization of on-line lightpath topology by listing all the connections present on the network and tracing the lightpath route taken by each of these connections on the physical network topology. These connections that are currently in progress define the network state.

2.3 Trunk Switched Network (TSN) and MICRON Framework

ISTOS supports modeling of both homogeneous and heterogeneous WDM grooming networks with respect to the nodes' grooming and switching architectures. ISTOS employs a general network model for WDM grooming networks called Trunk Switched Networks (TSNs) [19] and a framework for connection establishment in heterogeneous WDM grooming networks called Methodology for Information Collection and Routing in Optical Networks (MICRON) [18].

2.3.1 Trunk Switched Network (TSN)

A TSN is a two-level network model in which a link is viewed as a set of channels and a node views the link as a set of trunks, which are groups of channels with identical characteristics with respect to the capability of that node. The TSN is capable of modeling a network with sub-wavelength level traffic grooming over a link.

A TSN consists of nodes interconnected by links. Each link contains a set of channels. Channels in a link can be represented by a 4-tuple, (l, f, w, t) that describes the link, fiber, wavelength, and time-slot identifiers of the channel. Figure 2.1 illustrates the layered representation of the channel. A channel is the minimum granularity of bandwidth that is accessible on a link.

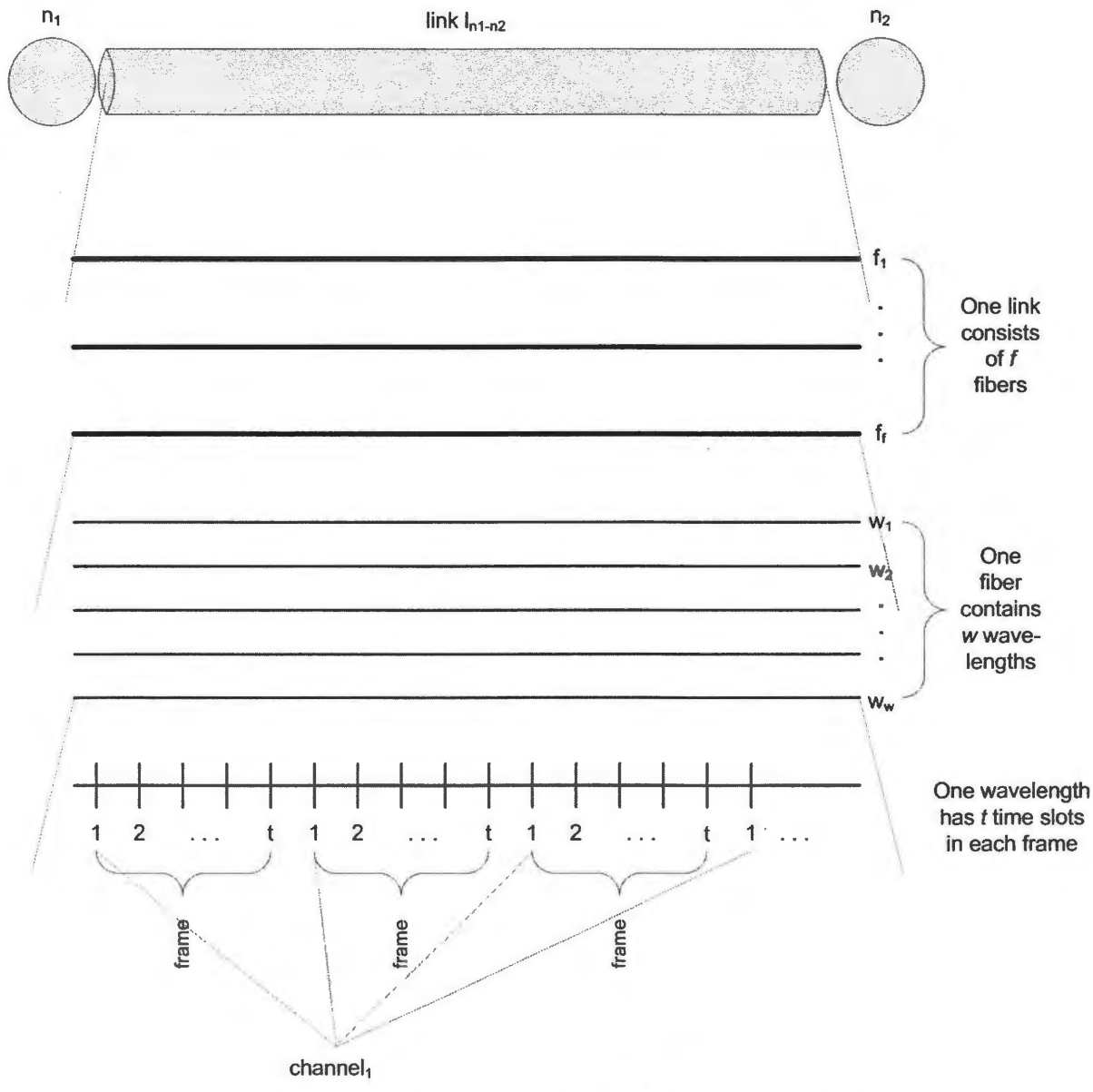


Figure 2.1. Link representation in TSN network.

Besides being a source or a destination of a connection in a network, a node can also act as an intermediate node for other connections that pass through it. Therefore, the function of a node includes switching of channels from one link to another link in order to facilitate a connection. Figure 2.2 shows the switching architecture of a node in TSN. It shows a node with four links. The trunks from each link are first demultiplexed and each of the trunks is then fed into a full channel interchanger. Trunks from different links are fed into their respective trunk switches. The output trunks from trunk switches are fed into a final stage of

full channel interchangers similar to that employed in the input stage. The trunks from different switches are then combined using trunk multiplexers and sent out to their corresponding output links. One input link and one output link in the switch are dedicated for the node to source and sink its own traffic. The full channel interchanger allows connection on a channel on a link to be switched to any channel within the same trunk.

The definition of a trunk is only with respect to a node. With respect to a link, the channels in the link are classified into groups called subtrunks. Two nodes connected physically by a link may view the channels in the link as different sets of trunks depending on the switching capability at the node. Sub-trunk xy of a link l_{ij} connecting nodes i and j, denoted by θ_{xy}^{ij}, are the set of channels that belong to both trunk x at node i and trunk y at node j.

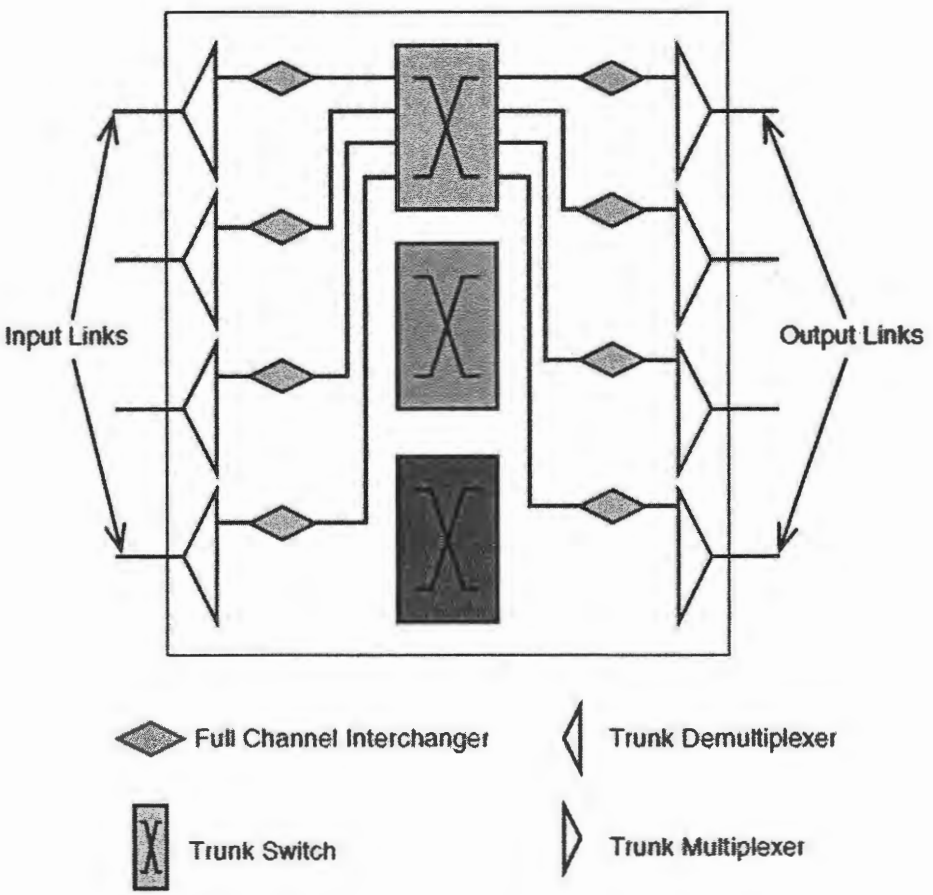


Figure 2.2. Node architecture in TSN network.

A TSN is said to be homogeneous if the collection of channels that constitute a trunk at a node is the same for all the other nodes in the network. Otherwise, it is heterogeneous.

To demonstrate how an optical network is modeled as a TSN network, we consider a WDM grooming network with the following links characteristics: two fibers per link, one band per fiber, three wavelengths per band, and two time slots per wavelength. The distinct fibers, wavelengths, and time-slots are denoted as f_1 and f_2; w_1, w_2, and w_3; and t_1 and t_2, respectively. We assume that fiber and band conversions are always possible.

1. Time-slot interchanger and wavelength conversion are not available on the network

   Each node views a link connected to it as six trunks in which each wavelength and time slot combination forms a trunk, and each trunk has two channels (one on each fiber). The set of trunks are: $\{(l, f, w_1, t_1), (l, f, w_1, t_2), (l, f, w_2, t_1), (l, f, w_2, t_2), (l, f, w_3, t_1), (l, f, w_3, t_2)\}$, where f can be f_1 or f_2 in each trunk.

2. Full wavelength conversion is employed, but no time slot interchange

   A link is viewed as two trunks, where each time slot on all wavelengths forms a trunk, and there are six channels on each trunk where each fiber and wavelength combination makes up a channel. The two trunks are: $\{(l, f, w, t_1), (l, f, w, t_2)\}$, where $f = f_1, f_2$ and $w = w_1, w_2, w_3$.

3. Only time-slot interchange is employed

   There are three trunks in each node, where each wavelength forms a trunk, and there are four channels per trunk, where each fiber and time slot combination forms a channel. The set of trunks are: $\{(l, f, w_1, t), (l, f, w_2, t), (l, f, w_3, t)\}$, where $f = f_1, f_2$ and $t = t_1, t_2$.

4. Both time slot and wavelength conversions are available

   In this case, the entire link is treated as one trunk with twelve channels, made up by fiber, band, wavelength, and time-slot combinations. Such a node is referred as a *full grooming node*.

TSN network model cannot effectively model a wavelength converter with limited conversion. Thus, only full wavelength conversion can be modeled in ISTOS. ISTOS also assumes utilization of *full-permutation trunk switches* on every node; any free channel at any input link can be switched to any free channel at the output link within the same trunk. A channel on a link is said to be busy if it is allocated for a connection; otherwise, it is said to be free.

By utilizing this generalized framework, ISTOS is able to model both homogeneous and heterogeneous networks. The GUI performs the translation from user inputs to the TSN input parameters.

2.3.2 MICRON Framework

ISTOS is built based on MICRON framework [18], a framework that specifies the information needs to be collected and processed in order to establish connections in WDM grooming networks with heterogeneous switching architectures.

Connection establishment in a circuit-switched network consists of two steps: path selection and channel assignment. In the *path selection* step, a path is selected based on certain criteria to route the connection from source to destination. One or more channels, depending on the connection requirement, are then assigned to each link that the chosen path encompasses during *channel assignment* phase. Connection establishment in a TSN network, however, consists of three steps: selecting path, assigning subtrunk on every link along the selected path, and reserving channel(s) on every subtrunk on every link.

MICRON framework specifies the type of information collected on each link and how this information can be combined to obtain the path information that is useful in path selection and channel/subtrunk assignment process.

Link and Path Information Matrices

The numbers of available channels on every subtrunk of that link are represented in a matrix called link information matrix stored at each link. The *link information matrix* (LIM) of a link connecting node i to node j is denoted $L_{ij} = [l_{xy}]$ and represented by a K_i x K_j matrix, where K_i and K_j are the number of trunks at nodes i and j respectively, and l_{xy} represents the number of free channels in subtrunk θ_{xy}^{ij}.

Matrices of links that span a path are combined to gain information on that path. The *path information matrix* (PIM) from node i to node k, denoted by P_{ik}, through node j is obtained by multiplying the matrices of individual path segments P_{ij} and P_{jk}. PIM of a single-hop path is equivalent to the link information matrix of the link that path spans. Path information matrix indicating maximum available channel that can be routed on the path that starts at a certain trunk x at source node and ends at trunk y at destination node can be obtained by applying a set of $(\min, \max)$ operations when multiplying the corresponding link information matrices. Depending on the context, the typical operators $(\times, +)$ might be used instead.

The path information matrix is further compacted into a path information vector (PIV) to minimize amount of information transferred between nodes. PIV of a node k for a path p with source i and destination k, denoted by V_{ik} of dimension $1 \times K_k$, is obtained by multiplying PIV at node i and P_{ik}. The PIV at source node i, denoted by U_i, is always set as a unit row vector of size K_i.

To illustrate how link information matrix and path information matrix are established, we consider a simplified version of network in Figure 1.1. Each link in the network has two fibers per link, one band per fiber, three wavelengths per band, and two time-slots per wavelength. We consider the following configuration as illustrated in Figure 2.3: node 1 has wavelength converter, but not time-slot interchanger; node 2 has neither type of converter; node 3 and 4 have both the time-slot and wavelength converters, and node 5 and 6 have only the time-slot interchanger.

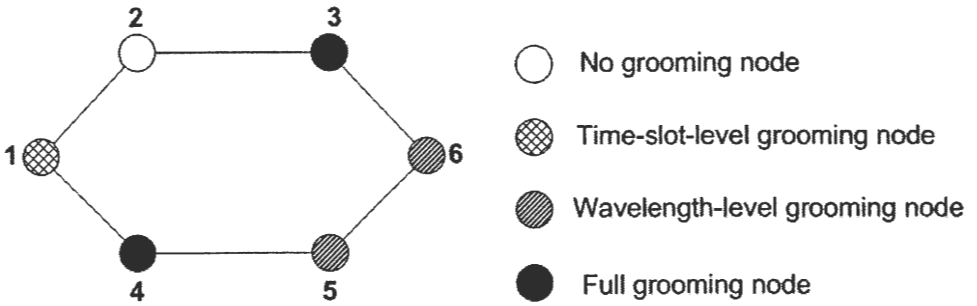


Figure 2.3. An example heterogeneous WDM grooming network.

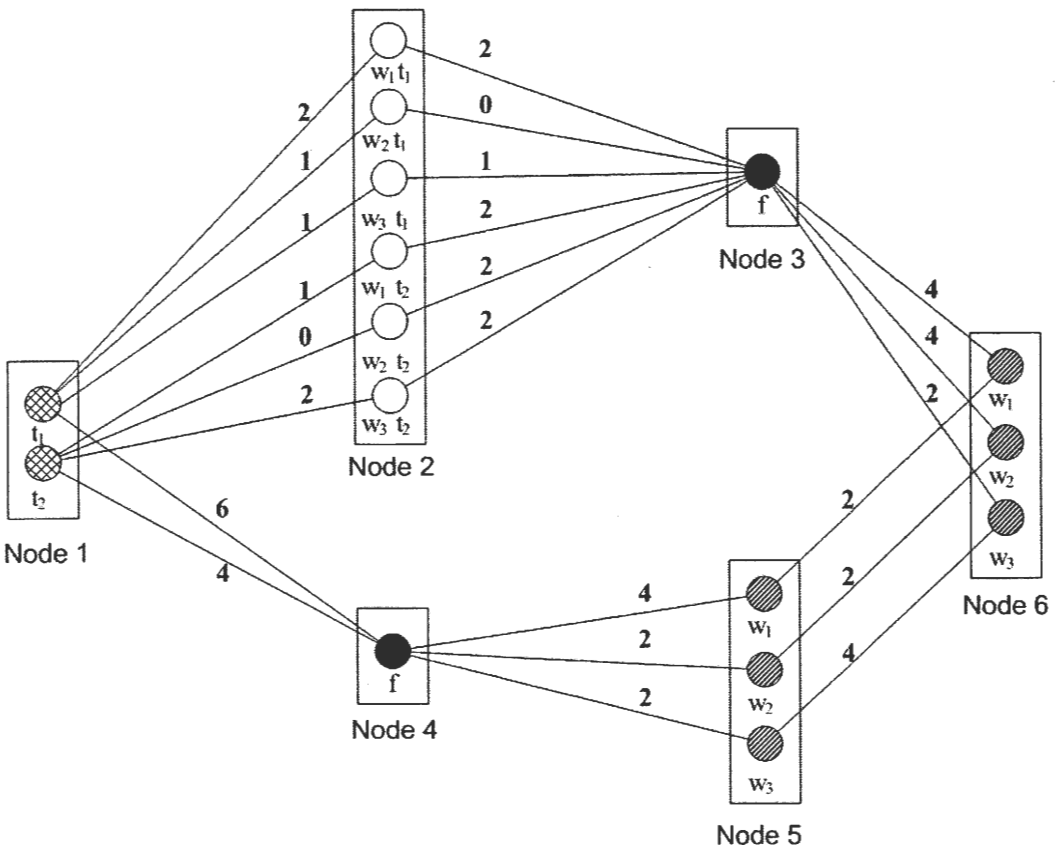


Figure 2.4. Expanded view of network in Figure 2.3 with free capacity information.

Figure 2.4 shows the expanded view of the same network illustrating how the links are observed as trunks and subtrunks in TSN. Time-slot-level grooming nodes view the link as 2 trunks, denoted by t_1 and t_2, with 6 channels in each trunk, while wavelength-level grooming nodes view the link as 3 trunks, denoted by w_1, w_2, and w_3, with 4 channels in each. Full grooming nodes view a link as one trunk, denoted by f, with 12 channels. Nodes with no grooming capability view the link as 6 trunks, denoted by $w_1\,t_1$, $w_2\,t_1$, $w_3\,t_1$, $w_1\,t_2$, $w_2\,t_2$ and w_3

t_2, with two fibers representing the channels in each trunk. The number indicated on each subtrunk of the links represents the free channel capacity available on the subtrunk. The maximum capacity of each subtrunk θ_{xy}^{ij} can be determined by finding the channels that are common in both trunk x at node i and trunk y at node j.

The link information matrix indicating the subtrunks capacity of each link is listed in Figure 2.5. The path information matrix on path 1-2-3-6 calculated on operators $(\times,+)$ is:

$$P_{1-6} = P_{1-2} \cdot P_{2-6} = L_{1-2} \cdot P_{2-3} \cdot P_{3-6} = L_{1-2} \cdot L_{2-3} \cdot L_{3-6} = \begin{bmatrix} 20 & 20 & 10 \\ 24 & 24 & 12 \end{bmatrix}$$

The element p_{xy} of the matrix P_{1-6} indicates the number of possible channel assignment combinations for a connection with one channel capacity requirement that starts at trunk x at node 1 and ends at trunk y at node 6.

The path information matrix for the same path that is calculated using operators $(\min,\max)$ is:

$$P_{1-6} = \begin{bmatrix} 2 & 2 & 2 \\ 2 & 2 & 2 \end{bmatrix}$$

An element p_{xy} of the above matrix denotes the maximum capacity that can be routed from trunk x at node 1 and ends at trunk y at node 6 without splitting the connection.

$$L_{1-2} = \begin{bmatrix} 2 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 2 \end{bmatrix} \qquad L_{3-6} = \begin{bmatrix} 4 & 4 & 2 \end{bmatrix}$$

$$L_{1-4} = \begin{bmatrix} 6 \\ 4 \end{bmatrix}$$

$$L_{2-3} = \begin{bmatrix} 2 \\ 0 \\ 1 \\ 2 \\ 2 \\ 2 \end{bmatrix} \qquad L_{4-5} = \begin{bmatrix} 4 & 2 & 2 \end{bmatrix}$$

$$L_{5-6} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 4 \end{bmatrix}$$

Figure 2.5. Link capacity matrices corresponding to Figure 2.4.

Connection Establishment

The connection establishment is carried out in two passes: forward and reverse passes. During the forward pass, each node on the candidate paths forwards its path information vector (PIV) and the connection request to the next node along the path. The destination node may then use the collected path information vectors as criteria for selecting a path for the connection request. Different routing algorithms employ different techniques in utilizing this information for choosing a suitable path.

During the reverse path of the connection establishment, a subtrunk on every link along the selected path is chosen, starting at the destination node and ending at the source node. Connection is established by reserving enough capacity required by that connection on selected subtrunks. The path information vector available at each node may be used to assist in subtrunk selection process.

2.4 Routing, Trunk Assignment and Network Survivability

In dynamic-trafficked optical networks, a lightpath is established for each connection request as it arrives. The routing and trunk assignment protocols determine what factors go into consideration when selecting path to route connection from source node to destination node and the policy employed for choosing the subtrunks on each link on the prior-chosen path, respectively. Robust networks are also designed to account for failures; such networks are armed with protection schemes to survive failures and so that reconfiguration can be done to minimize the number of failed connections in case of failure.

ISTOS currently incorporates several routing and channel/trunk assignment algorithms and network protection schemes in its algorithms library. The routing algorithms included are: fixed alternate path, widest shortest path, shortest widest path, maximum sum shortest path, shortest maximum sum path, shortest path on hop count, maximum shortest path, shortest

maximum path, and available shortest path, and the trunk assignment is either random, first-fit and best-fit trunk assignment heuristic.

The network protection algorithms implemented are $(+1)$ sub-graph approach, dedicated backup, backup multiplexing, failure-dependent path protection, connection-switched link protection, and diversion.

2.4.1 Routing Schemes

The routing schemes included in ISTOS can be categorized into two types: static and dynamic routing.

Static routing. The routing in static scheme is performed off-line; the scheme presumes the existence of pre-calculated path(s) between each pair of nodes in the network. Although connection requests are generated dynamically based on Poisson traffic distribution in ISTOS, the set of pre-determined paths taken into consideration in the static routing algorithm remain the same. The network state might be taken into account when the algorithm needs to choose the route among pre-selected '*k*' paths.

Dynamic routing. Dynamic routing scheme, on the other hand, chooses the route from a source to a destination node dynamically, depending on the network state in progress.

Fixed alternate path. Of all the routing protocols included in ISTOS, only the fixed alternate path algorithm falls into static routing category. This algorithm chooses the first available path out of the pre-determined '*k*' candidate paths in specified order.

When this algorithm is selected as the primary or backup routing metric in ISTOS simulation, a separate file indicating the set of paths for each pair of nodes in considered network has to be provided. This file has to be in the format as described in Section 3.2.

Widest shortest path. The widest shortest path routing (WSP) [17] scheme looks into the path information vectors of all candidate paths and select the path with the widest ordered vector. These path information vectors denote the maximum available channel capacity that can be routed to every trunk at the destination node on each path. The vectors are each ordered in descending values of the individual channel capacity and then compared to get the largest path vector. An ordered vector $A=(A_1, A_2, \cdots, A_t)$ is said to be larger than another ordered vector $B=(B_1, B_2, \cdots, B_t)$ if for some $i\,(1 \le i \le t)$, $A_i > B_i$ and for all $j < i$, $A_j = B_j$. The path with largest path vector is said to be the widest path and is chosen to accommodate the connection request. In case of a tie, shortest path with minimum hop length is selected.

Shortest widest path. The shortest widest path [17] is the opposite of the widest shortest path algorithm. It chooses the path with least hop length, and in case of a tie, path with widest path vector is selected.

Maximum sum shortest path and shortest maximum sum path. The maximum sum shortest path algorithm first looks into the sum of available trunks capacity identified by the path information vector of each candidate path. Path with the maximum capacity sum is selected. In case of a tie, the shortest hop-length path is chosen. On the other hand, the shortest maximum sum path looks into the length of the path first, and only considers the path capacity sum as a tie breaker.

Shortest path on hop count. The shortest path on hop count is similar to the conventional Dijkstra's shortest path routing algorithm [25] based on hop-length.

Maximum shortest path and shortest maximum path. The maximum shortest path first identifies the maximum value in the path information vector of each candidate paths. This value represents the maximum capacity that can be routed on each path without splitting the connection. Path with the highest maximum capacity value is selected, and if more than one such path is available, path with shortest hop count is selected. The idea behind selecting

path with largest maximum available capacity is that the path is most probably least loaded and hence attempting to balance the network load. It also increases the probability of successful connection establishment on the selected path since we chooses a path with the highest possible channel capacity. The shortest maximum path algorithm selects a path in the reverse order as the maximum shortest path routing scheme.

Available shortest path. The available shortest path (ASP) [17] algorithm first identifies paths that are available among all those considered. Path with at least one non-zero element in its PIV is said to be available. ASP then chooses the shortest path among the available paths. If there are more than one paths with same length, the result would be selected randomly.

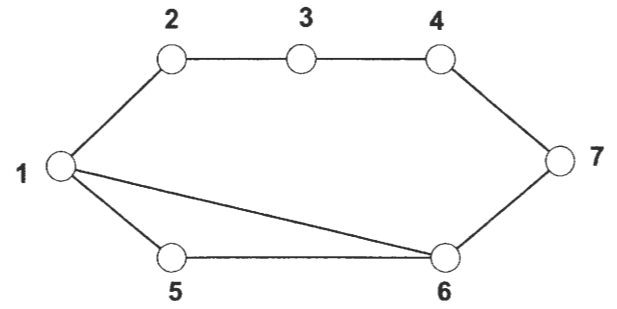
An example network to show how these algorithms select its path is shown in Figure 2.6. The operator set $(\min, \max)$ is used to calculate the path information matrices for the following possible paths: 1-2-3-4-7, 1-6-7, and 1-5-6-7, to route connection from node 1 to 7, and they are: $P_{1\text{-}2\text{-}3\text{-}4\text{-}7} = \begin{bmatrix} 2 & 2 & 2 \\ 2 & 2 & 2 \end{bmatrix}, P_{1\text{-}5\text{-}6\text{-}7} = \begin{bmatrix} 2 & 0 & 2 \\ 2 & 0 & 2 \end{bmatrix}, P_{1\text{-}6\text{-}7} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$.

The path information vector for each path is obtained from its path information matrix by getting the maximum capacity that ends at each trunk at the destination node. The shortest hop path is for obvious reason, the path 1-6-7. All algorithms that consider path with shortest hop first would also return this path as it is the only path that takes two hops to reach node 7. The widest shortest path routing algorithm selects path 1-2-3-4-7 because it has the largest ordered path information vector compared to the other two. Their corresponding PIVs are:

$$U_{1\text{-}2\text{-}3\text{-}4\text{-}7} = [2 \;\; 2 \;\; 2], U_{1\text{-}5\text{-}6\text{-}7} = [2 \;\; 0 \;\; 2], U_{1\text{-}6\text{-}7} = [1 \;\; 0 \;\; 1].$$

The maximum sum shortest path would also select the same path for it has the highest PIV sum, while the maximum shortest path would select path 1-5-6-7 because although path 1-2-3-4-7 has the same maximum capacity, its route is longer.

All three paths considered are available because each path PIV has at least one non-zero element. The available shortest path would then select the path 1-6-7 since it is the shortest among all.



$$L_{1-2} = \begin{bmatrix} 2 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 2 \end{bmatrix} \qquad L_{4-7} = \begin{bmatrix} 0 & 2 & 2 \\ 2 & 1 & 1 \end{bmatrix}$$

$$L_{1-6} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$$L_{2-3} = \begin{bmatrix} 2 \\ 0 \\ 1 \\ 2 \\ 2 \\ 2 \end{bmatrix} \qquad L_{1-5} = \begin{bmatrix} 6 \\ 4 \end{bmatrix}$$

$$L_{5-6} = \begin{bmatrix} 4 & 0 & 2 \end{bmatrix}$$

$$L_{3-4} = \begin{bmatrix} 3 & 6 \end{bmatrix} \qquad L_{6-7} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 4 \end{bmatrix}$$

Figure 2.6. An example network with its corresponding link information matrices denoting free channel capacity at each trunk.

The previously described routing protocols are all destination-specific routing schemes. In this approach, the protocol selects the best possible path to route the connection without the knowledge of the request characteristic, specifically the requested capacity. This way, the path selection protocol can be totally separated from the trunk assignment algorithms, and hence might be simpler than a combined routing and channel assignment algorithm. On the other hand, paths selected by the routing algorithm might not be able to accommodate the connection since requested channel capacity is not taken into consideration when selecting the path.

For this reason, ISTOS allows users to specify more than one routing schemes for each network. For every connection request, connection is attempted to be established using the first routing/path selection algorithm. If path selected is not available, i.e. not able to accommodate the connection, the second routing algorithm is used and connection establishment is attempted again. This process repeats on the next routing scheme, in

sequence listed by user, until connection is established. If none of the routing scheme succeeds, the connection is said to be blocked.

2.4.2 Trunk Assignment Algorithms

After a path is selected, trunk assignment policy is used to select trunk at each node along the route. Since ISTOS employs the MICRON framework, the trunk selection is done on the reverse direction of the path establishment. This means the trunk at the destination node is first selected and next is the previous node along the path and so on.

The three trunk assignment algorithms included in ISTOS are: random, first-fit, and best-fit assignment. All three algorithms utilize the path information vector (PIV) that has been calculated on each node along the chosen path. The trunk to end the connection at the destination node is selected using the PIV of that node. Trunk assignments at other nodes are made using the *ratio vector* [18].

Let i be the source node and x_k be the trunk chosen at node k. To select a sub-trunk on link $j-k$, a ratio vector is computed at node k. The ratio vector, denoted by R_{jk}, is obtained as the product of the path information vector at previous node, V_{ij}, and the transpose of the column vector of the link information matrix L_{jk} corresponding to x_k, i.e. $L_{jk}^T(x_k)$:

$$R_{jk} = V_{ij} \times L_{jk}^T(x_k) = \begin{bmatrix} v_1 & \dots & v_{K_j} \end{bmatrix} \circ \begin{bmatrix} l_{1x_k} & \dots & l_{K_j x_k} \end{bmatrix} = \begin{bmatrix} v_1 \circ l_{1x_k} & \dots & v_{K_j} \circ l_{K_j x_k} \end{bmatrix}$$

The operator $\circ$ denotes the element-wise operation on the row vectors. ISTOS uses the operator $\times$ to compute ratio vectors as corresponding to operator $(\times,+)$ in its path information matrix calculation. This means the trunk assignment algorithm selects a channel out of the set of all possible channel assignments possible on the path.

Random trunk assignment. The random trunk assignment heuristic chooses a trunk in random among the subtrunks that have at least one free channel capacity. Trunk x at node j is

chosen with a probability of $\frac{e_x}{\sum_{y=1}^{K_j} e_y}$, where e_x denotes the element r_x of ratio vector R_j or v_x of PIV V_j if j is destination node, and K_j denotes the number of trunks at node j. This means that subtrunks with higher capacities have higher probabilities of being selected. This way the algorithm increases the probability of the selected subtrunk to accommodate the requested capacity.

First-fit trunk assignment. The first-fit heuristic chooses the first available trunk. Trunks at each node are assumed to be in numerical order, and trunks are tested for its availability in that order. This is similar to the first-fit wavelength assignment algorithm [7].

Best-fit trunk assignment. The best-fit trunk assignment algorithm chooses subtrunk with the least remaining capacity after requested capacity is allocated. This scheme adopts the same concept as best-fit wavelength assignment policy to establish connection on trunk that can just accommodate the capacity of the request in order to increase the probability of accommodating the upcoming traffic.

Channels within a subtrunk are not differentiable in ISTOS. To simulate a connection establishment, exact channel capacity requested by the connection call is taken away from the total available channel capacity of the chosen subtrunk on each link on the selected path. It can thus be said that the channel assigned to a connection is selected randomly from the set of available channels in the selected subtrunk on each link on the chosen route.

Routing as well as trunk assignment algorithms for primary and backup paths are specified separately in ISTOS. The combination provides a wide array of simulation parameters in a network that directly determine its performance.

2.4.3 Network Survivability Strategies

Network survivability techniques can be categorized into either path or link protection scheme depending on how backup path is routed. ISTOS covers four path protection schemes: (+1) subgraph routing, dedicated backup, backup multiplexing, and failure-dependent path protection (FDPP), one link protection strategy: connection-switched link protection (CSLP), and one combined protection scheme called the diversion strategy.

(+1) Subgraph Routing

The (L+1) fault tolerant technique [11] guarantees that network can be reconfigured to a state in which all connections will remain established in the presence of a single link failure. This is done through associating each network with a set of subgraphs, each of which represents the original network with one of its links removed. A network with L links has a set of L subgraphs, and so a total of (L+1) network states have to be maintained simultaneously. Hence the naming of (L+1) subgraph routing strategy.

In this strategy, a connection request arriving on the network is accepted if and only if it can be routed in the original network and all its subgraphs. The primary routing and trunk assignment metrics specified by ISTOS user are used to find the request's route in original network, while the backup metrics are used to find routes in the subgraphs. In the event of a single link failure, the network is restored to the state of the corresponding subgraph in which that failed link is removed, where all connections are guaranteed restoration. Note that the (L+1) strategy can only handle single link failures.

The (L+1) subgraph routing can be extended into a generalized (+1) subgraph routing in which the set of subgraphs maintained for network operation corresponds to the presence of

Shared-Risk Link Group (SRLG) failures [13]. Each subgraph represents the original network with links belonging to an SRLG removed.

The (+1) subgraph routing strategy can be used to evaluate a network performance in the presence of single node failures. This is done by grouping the incoming and outgoing links connected to each node in the network into one SRLG group. Thus, a network with N nodes would have to maintain (N+1) network states.

Dedicated Backup and Backup Multiplexing

Dedicated backup and backup multiplexing are only effective against single link failures. In both techniques, a link-disjoint backup path is established for every connection request after its primary path is set up. This is done by removing the links on the chosen primary path before running the backup routing and trunk assignment algorithms. The request is accepted only if both the primary as well as backup path and subtrunks are found. In dedicated backup, no sharing of network resources is allowed. Backup multiplexing method allows backup paths whose primary paths are link-disjoint to share network resources. This results in more effective network resource utilization.

Failure Dependent Path Protection

Failure-dependent path protection (FDPP) scheme [23] attempts to outperform the (L+1) protection strategy high cost of reconfiguration by making sure only working or primary connections that are affected by a particular SRLG failure are reconfigured. This is achieved by having the connection aware of the set of SRLG failures that would result in its failure, denoted Ψ_C and $\Psi_C \subseteq \Psi$, where Ψ is the set of all possible SRLG failures in the network.

In FDPP, after a primary path is established, the backup path for each SRLG failure in Ψ_C is attempted. Only those connection requests whose primary and backup paths for all

SRLG failures in Ψ_C are established are accepted. Otherwise, the request is rejected or blocked.

To do this, FDPP keeps track of three additional information matrices for each link $l \in L$ besides the static S_l matrix, which indicates the total channel capacity in each subtrunk θ_{xy}^{l} on link l. They are:

- P_l: each element of the matrix, e_{xy}, represents the number of channels occupied by primary connections in subtrunk θ_{xy}^{l}.
- G_l^{ψ}: e_{xy} denotes the number of channels in subtrunk θ_{xy}^{l} that are occupied by primary connections that would become available due to failed connections corresponding to SRLG failure $\psi \in \Psi$.
- B_l^{ψ}: e_{xy} indicates the number of channels in subtrunk θ_{xy}^{l} used by backup paths need to be established corresponding to failure $\psi \in \Psi$.

Each of these matrices is updated everytime after a connection is accepted and they are applied to obtain the channel availability information needed to establish the primary and backup paths. The channel availability matrix for link l, denoted A_l, used to help in selection of primary path is found by taking the minimum number of free channels on the link considering all failure scenarios, including no failure scenario, i.e.

$$A_l = S_l - P_l - \max\left(0, \max_{\psi \in \Psi}\left[B_l^{\psi} - G_l^{\psi}\right]\right) \quad (1)$$

Before selecting backup path for SRLG failure ψ, A_l is updated to reflect the number of free channels in case of ψ occurs. Therefore,

$$A_l = S_l - P_l - B_l^{\psi} + G_l^{\psi} \quad (2)$$

Note that the difference is due to the fact that before primary path is established, we do not know which SRLG failures would affect the connection and so we take into consideration all possible failure scenarios. On the other hand, when backup path is to be established, we know which specific failure scenario we are considering. Because we look at backup paths

for each SRLG failure separately, this strategy does backup multiplexing, both primary-backup and backup-backup multiplexing, inherently.

Connection Switched Link Protection

Connection-switched link protection or CSLP [9] is the only strategy in ISTOS that re-routes a failed connection around the failed link(s). This means on a failure, the source node is not necessarily aware of the rerouted path unless the link connected to it fails. This reduces the amount of signaling needed for network reconfiguration, and thus reduces reconfiguration time.

In CSLP strategy, every link $l \in L$ in the network is assumed to have a fixed pre-computed backup path, denoted Z_l^{ψ}, on which all the connections flowing on the link could be re-routed for every $\psi \in \Psi$ in the case when failure ψ occurs. In ISTOS, Z_l^{ψ} is obtained by finding the shortest path around link l after disabling all links that fail under ψ.

The matrices maintained on each link are the same as those in FDPP. Capacity on primary connections, however, is not released on any SRLG failure since the connection is re-routed along the backup path for the failed link(s). Hence, the availability matrix of link l cannot be smaller than the available capacity for primary path under no failure scenario. A_l used for selecting the primary path is therefore the same as in Equation (1).

For each connection request arriving on the network, a primary path P_C and its subtrunks are established. While assigning the subtrunk on a link $l \in P_C$, the channel capacity on the backup path Z_l^{ψ} is taken into consideration to guarantee the consistency of subtrunk assignment along the backup path with the one on link l. In other words, the starting and ending trunks on the backup path Z_l^{ψ} have to be the same as the ones on link l. The availability matrix of link l is therefore updated to be:

$$A_l = \min\left\{S_l - P_l - \max\left(0, \max_{\psi \in \Psi}\left[B_l^{\psi} - G_l^{\psi}\right]\right), \min_{\psi \in \Psi_l} R_l^{\psi}\right\} \quad (3)$$

where $R_l^{\psi} = \prod_{l' \in Z_l^{\psi}} \left[S_{l'} - P_{l'} - B_{l'}^{\psi} + G_{l'}^{\psi} \right]$, Ψ is the set of SRLG failures in the network, and Ψ_l is a subset of Ψ under which l fails, i.e. $\Psi_l = \{\psi | \psi \cap l \neq \varnothing\}$.

From path P_C, the algorithm then computes ψ_C, the set of SRLG failures that affects the primary connection. If the connection requests for protection, a backup path for every $\psi \in \Psi_C$ is established. First, the subtrunk assignment is done on the backup path Z_l^{ψ} for each link $l \in P_C$ that fails on ψ by using the updated available capacity matrix of each link $l' \in Z_l^{\psi}$ as shown in Equation (2). The connection request is rejected if the subtrunk can not be obtained such that assignment starts and ends at the subtrunks assigned at the beginning and end nodes of link l, respectively. Lastly, the backup path for ψ, denoted P_C^{ψ}, and its subtrunks assignment are computed by replacing every link $l \in (P_C \cap \psi)$ with its Z_l^{ψ} and its subtrunks.

After primary and backup paths are established for a connection request, the link matrices that denote the primary capacity, the gained capacity under a failure $\psi \in \Psi$, and the backup capacity needed on ψ are then updated as below:

- For each link $l \in P_C$, the requested capacity, denoted C_R, is added to assigned subtrunk (x_l, y_l) of P_l.
- For every link $l \in P_C$ and failure $\psi \in \Psi_C$, C_R is added to assigned subtrunk (x_l, y_l) of G_l^{ψ}.
- C_R is added to the assigned subtrunk (x_l^{ψ}, y_l^{ψ}) of B_l^{ψ} for every link $l \in P_C^{\psi}$ and $\psi \in \Psi_C$.

Diversion

Diversion scheme [20] attempts to bring out both the benefits of high network utilization in path protection strategies and the fast connection recovery time in link protection strategies. It is similar to link protection as it keeps the non-failed connection route from source to node before the first failed link when failure occurs. However, instead of re-routing

each failed link to its neighbor node, it diverts the connection from the starting node of the first failed link directly to the destination node.

The diversion path for a link is determined dynamically. When a connection request C comes, the followings are done:

1. The diversion path, denoted $y_l^{\psi}(C)$, for each link $l \in \psi$ to the destination node, denoted d_C, is computed for every SRLG failure $\psi \in \Psi$.

   The availability matrix for each link l used to compute the path is the same as in Equation (2). We only takes into consideration connections that remain or are reassigned in case of failure ψ, and therefore, backup multiplexing are inherent in this scheme, just like in FDPP.

2. The primary path and its subtrunk assignment are attempted.

   As in CSLP, the subtrunks assigned for the primary path and all the backup paths have to be consistent. Thus, the available capacity matrix used to select primary path has to account for the channel availability under each failure scenario $\psi \in \Psi$, and it is similar to Equation (3) except that in this case, R_l^{ψ} is the channel availability matrix of the diversion path, i.e. $R_l^{\psi} = \prod_{l' \in y_l^{\psi}(C)} \left\lfloor S_{l'} - P_{l'} - B_{l'}^{\psi} + G_{l'}^{\psi} \right\rfloor$.

3. If a primary path and its subtrunks are successfully obtained, a backup path for every $\psi \in \Psi_C$ is then attempted. Otherwise, connection request is rejected and the subsequent steps are skipped.

   The backup path is computed by concatenating part of the primary path until the first failed link with the diversion path of that link. Since the diversion path has been determined in step 1, the backup path can always be found. The concern is then to find the subtrunk assignment such that it is consistent with the primary path. The availability matrix used in step 2 guarantees this consistency once subtrunks for primary path are assigned.

4. Now that the primary path and backup paths for all SRLG failures that affect the primary path and their corresponding subtrunks are assigned, the link capacity information has to be updated accordingly, just as in CSLP.

As described above, all the network survivability schemes reserve one or more backup paths for each connection request to counteract against failures. In ISTOS, the routing and trunk assignment algorithms used to find the backup route are specified separately from those used for primary path. This allows user to find the most efficient routing and trunk assignment algorithms that complements well with the protection protocol.

2.5 Summary

In this chapter, we presented the TSN network model and the MICRON framework that ISTOS employs to model and establish connections for the WDM grooming networks that it is analyzing. We illustrated how a WDM grooming network with various grooming architectures can be modeled into TSN. We also presented the routing, trunk assignment, and network protection algorithms that have been implemented in ISTOS.

CHAPTER 3 Simulation in ISTOS

ISTOS is concerned with both the design and operational issues arising in optical WDM grooming networks. Issues in optical network design in general include designing a lightpath topology that minimizes the total network cost, dimensioning the network to meet current and future traffic demands based on a projected traffic pattern, etc. The network operation involves configuring the network to handle a certain fixed traffic matrix or a dynamic traffic based on a certain projected traffic distribution, detecting failures and reconfiguring network to restore disrupted services, monitoring network for proper operation and performance measurement, etc. [16]

ISTOS provides simulation of dynamic traffic to solve both the problems of dimensioning heterogeneous WDM grooming networks and deciding on the operational policies employed in the network, which includes the routing and trunk assignment algorithms and the network protection or reconfiguration scheme.

3.1 Networks Comparison for Resource Dimensioning

A key aspect of a wavelength-routed network design is to determine the link capacity, in other words the number of wavelengths, on each link on the WDM network. This is known as the wavelength dimensioning problem [16].

ISTOS provides an efficient way to solve the wavelength dimensioning problem with the use of blocking traffic model [16]. In this model, connection requests are assumed to arrive and leave at random time according to a statistical traffic pattern. The rate of traffic arrival is assumed to be the same as its termination rate. Each request upon arrival attempts a path to

route the connection and the corresponding subtrunks assignment. If a path and the subtrunks are not successfully established, the request is said to be blocked. The goal here is then to dimension the WDM links such that the blocking probability is relatively low.

The wavelength dimensioning problem can be solved by running multiple networks differing by the link capacities on ISTOS. The blocking probabilities for a given set of capacities on each network can be calculated and thus, the minimum set of link capacities that could be employed to achieve an acceptable blocking probability can be determined.

Experiment Parameters

More than one networks can be simulated by ISTOS simultaneously. These networks are combined together to form an experiment. The only constraint for networks to be in one experiment is that they have to be of the same physical topology. In other words, these networks must have the same number of nodes, links, and same connectivity structure.

Networks within an experiment are simulated with the following same simulation parameters:

- Number of rounds and number of requests per round

  Connection requests are generated dynamically to evaluate the performance of each network in handling dynamic traffic. Metrics monitoring performance of each network are reset after each round. A round is completed when the number of requests generated reaches the user-specified number of requests per round value. The simulation stops after all specified rounds are generated.

  A network starts with zero established connections initially. Hence, the performance metric of the first round might not be that realistic. An option to drop the result of the first round is available in ISTOS.

- Traffic and fault specifications

  The traffic generated is the same on each network. Traffic is generated at a rate as specified by user. The maximum bandwidth granularity of each generated connection request is also specified by user.

  Failures, if enabled, can be injected periodically after every certain number of requests are generated, or randomly at a rate specified by user. If random failure is chosen, user would specify the interval time between failures in the Mean Time to Next Failure (MTNF) variable and the average length of time for a failure in the Mean Time to Repair (MTTR) variable. As in traffic generation, failure is generated based on Poisson distribution function and every generated failure is distributed to each network. Each network then runs its network reconfiguration module to respond to the failure event. To generate identical failures, fault specifications, i.e. type of fault occurrence – periodic or random – and its corresponding parameter(s) – fault period for periodic failure or MTNF and MTTR for random failure, fault type – physical or logical, and fault component – link, node, or SRLG, have to be common among all networks.

The granularity of connection requests in ISTOS is defined in units of channel, rather than in Hertz (Hz). Note that the resources specified as the parameters of a two-unidirectional link in ISTOS are available for each direction, while resources specified for a bidirectional link are the total resources that are shared by each directional link. The parameters include the link's physical properties, such as its bit error rate (BER), propagation delay and jitter generated, and the number of amplifiers and regenerators along the link.

Since ISTOS handles heterogeneous WDM grooming networks, there is another possible dimensioning problem that a network operator could solve. Given a limited number of converters, find the most effective resource distribution that minimizes the blocking probability of the dynamic traffic following a specific distribution.

To solve this problem, typically an iteration of the analysis of a network with specific setting of converters distribution has to be done. ISTOS simplifies this iteration step by

allowing concurrent simulation of multiple networks which differ by the node conversion types. In this case, the capacity of each link might be kept the same for all networks.

3.2 Algorithm Comparison

One of the many network operational issues concerns the decision as how the path to route a connection request is selected and what kind of policy is used to assign subtrunk on each of the links on the chosen path so that the dynamic traffic on a specific network topology is handled to its best. Another issue along the same line takes into account the possibility of failures occurring in the network. A network usually provides backup path for each connection to guarantee a certain degree of failure tolerance. Hence, the problem has now extended to deciding the path and subtrunk selection policies for backup paths.

ISTOS allows concurrent simulation of several networks differing by the routing and trunk assignment protocols and/or network protection algorithm. This offers effective comparison of multiple algorithms across similar network topologies in one simulation run.

Since the primary concern here is deciding on the network operational policies that minimizes the blocking probability of requests based on a projected statistical traffic and provides service guarantee to a specific degree, networks run in one experiment usually comprises of the same logical network structure. The properties of each node and link are the same for all the networks. The difference lies in each network simulation parameters, in which the routing metrics, trunk assignment policy, and backup or protection strategy are specified.

Default Network Topology

To maintain the network physical topology, ISTOS utilizes a default network topology panel on which topology design for all networks within an experiment is done. Modifications

to network structure can only be done on experiment level, not on network level, on the default network topology panel. As a result, all networks in an experiment have the same number of nodes and links and each node pair is connected by the same link.

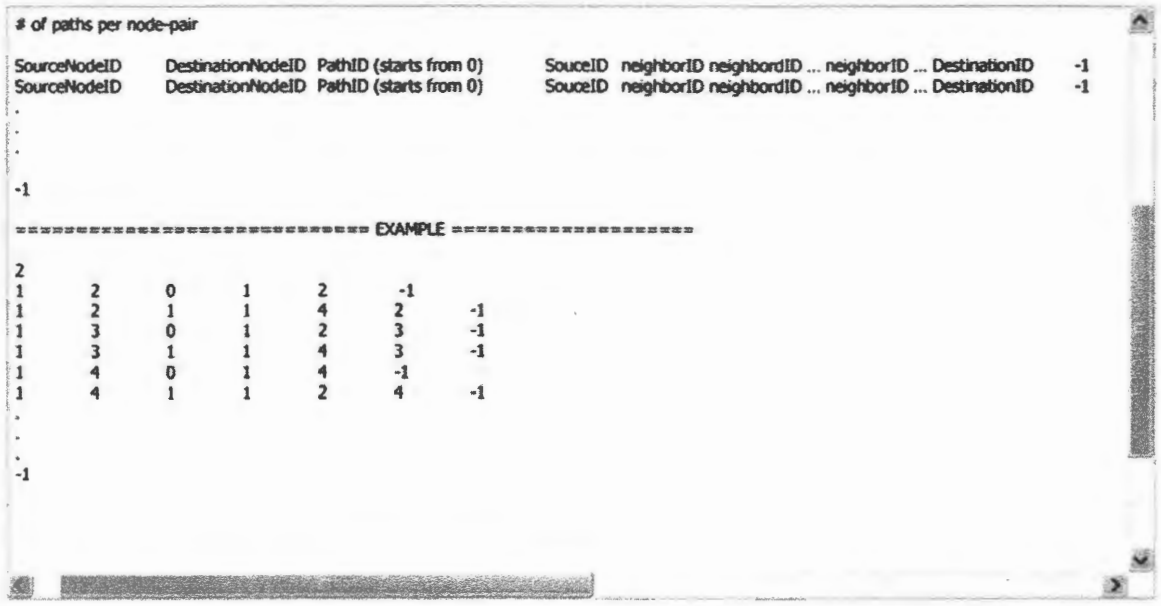
Node and link's properties can be propagated from the default network topology panel to other networks within the same experiment. This allows global specification of node and link parameters on all networks, which we might need for solving the network operational policies issue.

Routing Metrics

More than one algorithms can be specified in the primary and backup routing metrics. The algorithms are ordered by the sequence when the user enters them. During path establishment for a connection request, each of these algorithms is executed to attempt routing of the connection in order. The execution stops on the first algorithm that successfully establishes the connection route. The flexibility of specifying more than one routing algorithms increases the chance of connection requests being accepted and successfully routed.

An additional file has to be attached when fixed alternate path routing algorithm is selected. This file specifies the set of paths to be considered for the routing between each node pair. The format of this file is shown in Figure 3.1. An example of the path lists for the network shown in Figure 1.1 follows. The maximum number of paths that can be specified for each node pair is fixed at 4.

To evaluate the performance of an algorithm in response to different network load, an experiment can be run repeatedly with varied traffic arrival rate. This is to assess how well an algorithm adapts if traffic arrives at a rate as not expected but still follows the projected distribution pattern.



```
# of paths per node-pair

SourceNodeID      DestinationNodeID  PathID (starts from 0)      SouceID  neighborID neighbordID ... neighborID ... DestinationID     -1
SourceNodeID      DestinationNodeID  PathID (starts from 0)      SouceID  neighborID neighbordID ... neighborID ... DestinationID     -1
.
.
.

-1

============================== EXAMPLE ======================

2
1       2       0       1       2       -1
1       2       1       1       4       2       -1
1       3       0       1       2       3       -1
1       3       1       1       4       3       -1
1       4       0       1       4       -1
1       4       1       1       2       4       -1
.
.
.
-1
```

Figure 3.1. Specification file format for fixed alternate path (FAP) routing algorithm.

3.3 Summary

In this chapter, we discussed the type of issues and kind of simulations that ISTOS could solve and provides. We described the simulation parameters that are common among all the networks to be analyzed in ISTOS. We described how the network structure of all networks within an experiment is maintained through the default network topology panel. We also mentioned the different type of evaluations that ISTOS offers.

CHAPTER 4 Performance Metrics

The following performance metrics [11] are provided as simulation statistics at the end of ISTOS simulation for assessment of the efficiency and effectiveness of routing and trunk assignment algorithm, network protection strategy, or network resources distribution under investigation:

1. Blocking probability.
2. Average path length and average shortest path length.
3. Effective utilization and actual utilization.
4. Offered load.
5. Probability of path reassignment.
6. Failure recovery time.
7. Survivability guarantee.

Each of these metrics is defined mathematically and described further in Sections 4.1 through 4.7.

4.1 Blocking Probability

Blocking probability is the probability that a connection request entering the network is rejected. It is defined as the ratio of the number of blocked requests, denoted B, to the total

number of requests generated, denoted R, and is shown in Equation (4). A request is blocked if none of the paths selected by user-specified routing algorithm(s) has enough resources to route the connection.

$$P_{blocked} = \frac{B}{R} \tag{4}$$

The value $1 - P_{blocked}$ indicates the percentage of requests that will be assigned a connection when run on a particular network topology employing certain routing and trunk assignment policies.

4.2 Average Path Length and Average Shortest Path Length

The average path length is compared to the average shortest path length to get an idea of how effective a routing strategy performs compared to the extended Dijkstra's shortest path (EDSP) algorithm [17]. Both these metrics are measured in terms of the number of hops, or the number of links along the path.

The average path length metric measures the average length of path taken by the accepted connection requests. It is defined mathematically in Equation (5), where L_i = path length of request i, denoted R_i. This metric depicts the effectiveness of the routing strategy in finding longer routes to accommodate connections.

$$\overline{P} = \frac{\sum_{i=1}^{R-B} L_i}{R-B} \tag{5}$$

The average shortest path length metric measures the average shortest path length between the source and destination nodes of accepted connections. It is defined in Equation (6) where SPL_i = shortest path length of request R_i. Under low arrival rates, this metric reflects the characteristics of the network for a particular traffic arrival rate [20]. This is because blocking probability is very low for low traffic rates, and hence most requests are accepted and assigned a connection.

$$\overline{SP} = \frac{\sum_{i=1}^{R-B} SPL_i}{R-B} \tag{6}$$

This metric also depicts the fairness of a routing algorithm in treating connection requests. If the algorithm treats the connections in a fair manner, this metric should remain constant regardless of network load [20]. This is because the shortest path length between any two nodes in a network does not change with arrival rate. On the other hand, the value of this metric would decrease if the algorithm prefers shorter connections as network load increases.

4.3 Effective and Actual Network Utilization

Network utilization is an indication of how much of the network is used over the course of simulation and whether there are enough resources available to handle the traffic demands. It is computed by assigning an effective capacity requirement for a connection request. The effective capacity requirement for a request R_i is the minimum capacity required in the network to support the request. The shortest path length from the source to destination node of the request, denoted SPL_i as in Equation (6), is utilized for this purpose and therefore, the effective capacity requirement for a request R_i with capacity C_{R_i} is $C_{R_i} \times SPL_i$.

The effective network utilization, denoted U_{eff} and is defined in Equation (7), measures the minimum amount of network resources needed to service all accepted connection requests if they were routed along the shortest path, while the actual utilization, denoted U_{act} and defined in Equation (8), measures the actual resource capacity used to route all accepted connections. Combination of both these metrics is an indication of how effective the routing strategy utilizes the network resources.

$$U_{eff} = \sum_{i=1}^{R-B} C_{R_i} \times SPL_i \tag{7}$$

$$U_{act} = \sum_{i=1}^{R-B} C_{R_i} \times L_i \tag{8}$$

To define the network utilization at any instant of time within the simulation period, the above equations are normalized with respect to the simulation duration. The simulation

duration can be computed as a function of R/λ , where R is as in Equation (4) and λ is the request arrival rate. This value is upper-bounded by the total link capacity in the network, i.e. $\sum_{l=1}^{L} C_l$, where $C_l = \sum_{y=1}^{col} \sum_{x=1}^{row} S_l(x,y)$ is the sum capacity of all subtrunks of link l and S_l is the maximum channel capacity matrix of link l .

To further normalize the utilization metric so that it lies in between 0 and 1, the value $U \cdot \lambda / R$ is divided by the total network capacity value defined above.

4.4 Offered Load

The offered load metric is defined as the product of the request arrival rate per node and the average request duration [16]. The request arrival rate per node can be simply computed by dividing the traffic arrival rate of the network, λ, by the total number of nodes in network, denoted N . This is because the source node of each request entering the network is chosen with equal probability among all nodes. The offered load metric is denoted mathematically in Equation (9) with μ being the exponential distribution parameter which determines the mean holding time of 1.0 for requests.

$$Load_{off} = \frac{\lambda}{N} \cdot \frac{1}{\mu} \tag{9}$$

This metric is utilized as basis to simulations of different networks as it represents the amount of load placed on each node in the network at any given time.

4.5 Probability of Path Reassignment

To guarantee valid service of each connection request during its presence in the network, a network protection scheme sets aside one or more backup paths when attempting to route the request. Upon recovery in case of failure, the failed connections are reassigned to their corresponding backup paths.

The $(+1)$ fault tolerant scheme independently routes each incoming request in the base network and its subgraphs. Because of that, a connection request may need to be re-routed even though failure occurred does not affect the connection's primary path. Other protection schemes attempt to re-route only the failed connections during recovery.

The amount of path reassignment that takes place is quantified to compare the effectiveness of the protection strategies. The probability of path reassignment is computed for this purpose. It is the measure of probability that a connection is re-routed upon failure occurrence [11].

The probability of path reassignment is reflected by the average number of backup paths that are different from the primary path. For each accepted connection request, the total number of backup paths that are routed on different path or have different trunk assignments than its primary path is divided by the total number of possible failures in the network to get the probability of path reassignment for that request. The average of this value over all accepted requests are then calculated.

In general, the probability of path reassignment is calculated using the following equation:

$$P(\text{path reassigned on single SRLG failure}) = \frac{\sum_{i=1}^{R-B} \sum_{\psi \in \Psi} P^{\psi}(R_i)}{(R-B) \cdot N_{SRLG}},$$

where $P^{\psi}(R_i)$ is 1 if the backup path of request R_i for failure ψ is different from the primary path or 0 if otherwise, and N_{SRLG} is the total number of SRLG failure scenarios in the network which is equivalent to number of elements in set Ψ.

Note that even though link protection strategies and diversion scheme would most probably re-route only portion of the connection path, the probability of path reassignment might still be indifferent from those strategies which reassign path from the source node. The failure recovery time, however, should be different and is expected to be lower than path protection strategies since failure notification signal does not need to reach the source node before re-routing begins.

4.6 Failure Recovery Time

Failure recovery time [22] is computed to evaluate network protection strategy. It is the estimated measurement of the length of time required to recover a failed connection.

It is assumed that upon detection of a failure, nodes that are connected to the failed link broadcast a failure notification message to its neighboring nodes, and every other node, upon receiving the notification message, shall forward it to its neighbors and reconfigure its switches corresponding to the failure indicated in the message.

The time from the instant failure ψ occurs to the time a node n is notified about the failure is referred as the failure notification time, denoted T_n^{ψ}. This value depends on the failure detection time, denoted α, and the message propagation delay. α is assumed to be constant as the failure is assumed to be detected due to loss of a periodic monitoring packet transmitted over the control channel. The message propagation delay depends on link propagation delay and the packet processing time. The notification message is converted from the optical to electronic domain to be processed at every node. The electronic overhead time, denoted γ, incurred on every hop or link accounts for the time taken by sending node to prepare and transmit the notification packet on a link and the time its neighbor or receiving node process that packet. Let τ_l denote the propagation delay for link l. The hop delay seen by the failure notification message on link l is then $\tau_l + \gamma$.

On a failure ψ, both nodes, say $n_l(\psi)$ and $n_l'(\psi)$, connected to the failed link l would broadcast the failure notification message independently. Hence, a node n will be aware of the failure from the message that arrives first. The failure notification time for node n is estimated as shown in Equation (10), where $\Delta^{\psi}\left(n_1, n_2; \{\tau_l + \gamma\}\right)$ denotes the cost of the least-cost path from node n_1 to n_2 with $\{\tau_l + \gamma\}$ as the cost metric for links under failure ψ.

$$T_n^{\psi} = \alpha + \min\left[\Delta^{\psi}\left(n_l(\psi), n; \{\tau_l + \gamma\}\right), \Delta^{\psi}\left(n_l'(\psi), n; \{\tau_l + \gamma\}\right)\right] \tag{10}$$

The switch configuration time at any node in the network is assumed to be a constant β. Note that every node can always reconfigure its switches upon notification of failure because the backup path(s) and their corresponding subtrunks are computed during the establishment of the primary path for the connection. A node starts its switch reconfiguration as soon as it receives the failure notification message. The time at which the reconfiguration for failure ψ is completed at node n is therefore $R_n^{\psi} = T_n^{\psi} + \beta$. Again, the time duration is measured from the failure instant.

The failure recovery time for a connection is defined as the time duration from the last information received on the primary path to the first information received on the backup path. It is defined mathematically in Equation (11), where T_C^{ψ} denotes the recovery time of connection C under failure ψ, $F_C^{\psi}(n)$ denotes the earliest time by which connection C can cross node n on its backup path after failure ψ, $L_C^{\psi}(n)$ denotes the latest time by which connection C can cross node n on its primary path after failure ψ, and y_C^{ψ} is the last node on the backup path segment P_C^{ψ} that needs to be reconfigured.

$$T_C^{\psi} = F_C^{\psi}\left(y_C^{\psi}\right) - L_C^{\psi}\left(y_C^{\psi}\right) \tag{11}$$

Let x_C^{ψ} denote the first node on backup path of connection C under failure ψ, P_C^{ψ}, that needs reconfiguration and z_C^{ψ} denotes the first node on the *last surviving segment on its primary path* P_C. The last surviving segment is the longest segment of the primary path ending at the destination node of connection C on which no link fails under failure ψ. Note that the nodes x_C^{ψ} and y_C^{ψ} depend on the protection strategy.

Information from node z_C^{ψ} transmitted before failure ψ is still valid and in transit to the destination node on the last surviving segment on P_C. If t denotes the latest time by which the connection C crosses the immediate predecessor node of n on P_C, denoted by $\mathrm{Pred}(n, P_C)$, and the propagation delay on the link l connecting nodes n and $\mathrm{Pred}(n, P_C)$ is τ_l, then the latest time by which the connection crosses n is at $L_C^{\psi}(n) = \min\left(T_n^{\psi}, t + \tau_l\right)$. The recursive computation of this value is shown in Equation (12).

$$L_C^{\psi}(n)=\begin{cases}\min\left[T_n^{\psi}, L_C^{\psi}\left(\mathrm{Pred}(n,P_C)\right)+\Delta_{P_C}\left(\mathrm{Pred}(n,P_C),n;\{\tau_l\}\right)\right] & \text{if } n\neq z_C^{\psi}\\ T_n^{\psi} & \text{if } n=z_C^{\psi} \text{ and } \psi\cap P_C=\varnothing\\ 0 & \text{otherwise}\end{cases}$$

(12)

Note that if a failure does not affect a connection, z_C^{ψ} is the source node. If it does, only information that has crossed over node z_C^{ψ} at failure point has the potential to reach the destination. Therefore, the time $L_C^{\psi}\left(z_C^{\psi}\right)$ is taken to be 0 (the failure instant).

Upon receipt of failure notification, a node n starts reconfiguring its switches and completes it by time R_n^{ψ}. Only after the reconfiguration completes, can node n route connections along the backup path. The earliest time at which the backup connection can cross n is therefore $F_C^{\psi}(n)=\max\left(R_n^{\psi}, t+\tau_l\right)$ assuming t is the earliest time at which the connection crosses the immediate predecessor of node n on the backup path, denoted by $\mathrm{Pred}\left(n,P_C^{\psi}\right)$ after failure ψ and link l connects the nodes n and $\mathrm{Pred}\left(n,P_C^{\psi}\right)$. Equation (13) shows the recursive computation of the time $F_C^{\psi}(n)$ for all node n in between x_C^{ψ} and y_C^{ψ}.

$$F_C^{\psi}(n)=\begin{cases}\max\left[R_n^{\psi}, F_C^{\psi}\left(\mathrm{Pred}\left(n,P_C^{\psi}\right)\right)+\Delta_{P_C^{\psi}}\left(\mathrm{Pred}\left(n,P_C^{\psi}\right),n;\{\tau_l\}\right)\right] & \text{if } n\neq x_C^{\psi}\\ R_n^{\psi} & \text{if } n=x_C^{\psi}\end{cases} \quad (13)$$

The average of the worst-case failure recovery time of connections within one simulation round is computed for each network in ISTOS. The worst-case failure recovery time for a connection is computed as the maximum recovery time among all failures which affect the connection. The corresponding standard deviation is also computed.

4.7 Survivability Guarantee

The survivability guarantee metric reflects the performance of a network survivability strategy. It shows the average percentage of connections that will survive in case of failures. It measures the ratio of total number of connections that can be re-established in the event of

a failure with respect to the number of connections present on the network right before failure occurs.

Most protection schemes attempt to achieve 100% survivability guarantee by dropping the connection if backup paths and their corresponding subtrunks could not be successfully found during connection establishment phase. This metric, however, is informative when used together with the blocking probability and path reassignment probability metrics to assess effectiveness and feasibility of a protection scheme.

4.8 Summary

In this chapter, we presented the metrics that ISTOS provides to assess the performance of different networks employing different routing, trunk assignment, and/or network protection strategies. These metrics are helpful in evaluating the effectiveness and feasibility of an algorithm in a specific WDM grooming network.

CHAPTER 5 ISTOS Architecture

ISTOS is composed of two main parts: a front-end Graphical User Interface (GUI) on which user draws the network topologies to be run in the simulation and specifies all necessary simulation parameters and where the simulation output is presented, and the back-end engine which acts as the simulation core that runs the dynamic network traffic simulation. The two ends are interconnected via a customized socket object named *IstosComm*.

The front-end is developed in C# [26] on .NET [12] environment, and hence can only run on Windows. The back-end, on the other hand, is developed in C/C++ and runs on both Windows and Linux. The front-end is connected to the back-end either via interprocess communication if the latter is run on the same Windows machine as the front-end, or via SSH or Secure SHell protocol if it runs on Linux.

Figure 5.1 describes the architecture of ISTOS showing the components of the front and back ends and how they communicate with each other.

5.1 Front-End Graphical User Interface

The GUI primarily comprises of two components: the main form and the output window. User builds the network topologies and establishes their corresponding specifications on the main form. The GUI then translates the topologies and specifications into configuration files to be sent to backend as inputs to the simulation. This translation process is transparent to users.

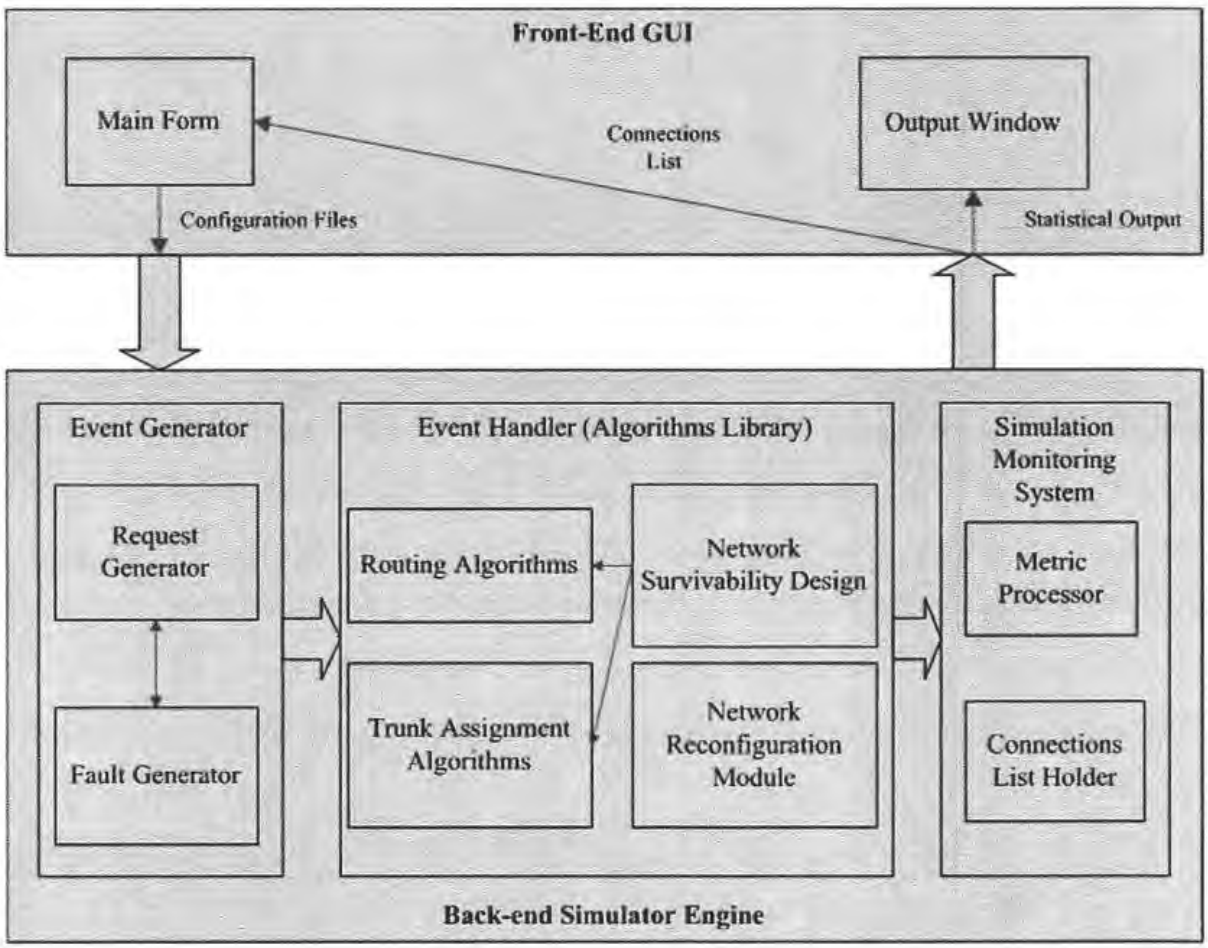


Figure 5.1. ISTOS architecture.

The configuration files transmitted as inputs to the back-end engine are comprised of two different files: the experiment and the network data files. The experiment file contains the simulation parameters that belong to all the networks, which are specified as the experiment's properties as described in Section 3.1, as well as each network's simulation parameters, which cover the user-chosen routing and trunk assignment policies, whether the network is equipped with a protection strategy, and whether truncation is employed. The network data file contains the network structure and the nodes' and links' properties. One network data file is produced for every network in the simulated experiment.

Output file(s) produced and transmitted by backend at the end of simulation is displayed on the output window. During simulation, list of connections currently established on every network is received and displayed on the *Connection Requests window* on the main form.

5.1.1 Main Form and Output Window

Main Form

The main form contains the following window components:

- Designer panel

  Instead of writing configuration files to run simulation, ISTOS front end offers an efficient graphical way for user to specify the network topology. An intuitive design panel, as illustrated in Figure 5.2, is provided for user to draw the topology of the network(s) to be simulated.

  Drawing toolbar can be found on top of the design panel and it provides easy access to node, link, and cursor buttons. A click on these buttons will generate the creation of node, link, and the return of cursor respectively.

  Copy and paste of selected objects (nodes or links) is available from right-click pop-up menu on the design panel. An option to propagate an object's property to selected networks within the current experiment is also available.

- Experiment explorer

  ISTOS allows creation and opening of multiple experiments in one application window. The currently opened experiments and the network topologies they contain are listed in the experiment explorer window in a tree format structure, as shown in Figure 5.3. Each experiment and network has its corresponding designer panel tab. When a tab is selected, its corresponding node in the experiment tree is selected, and vice versa.

  When user right clicks on a node in the experiment tree, the node's pop-up menu is displayed. User may add or remove a network topology, select to display an experiment

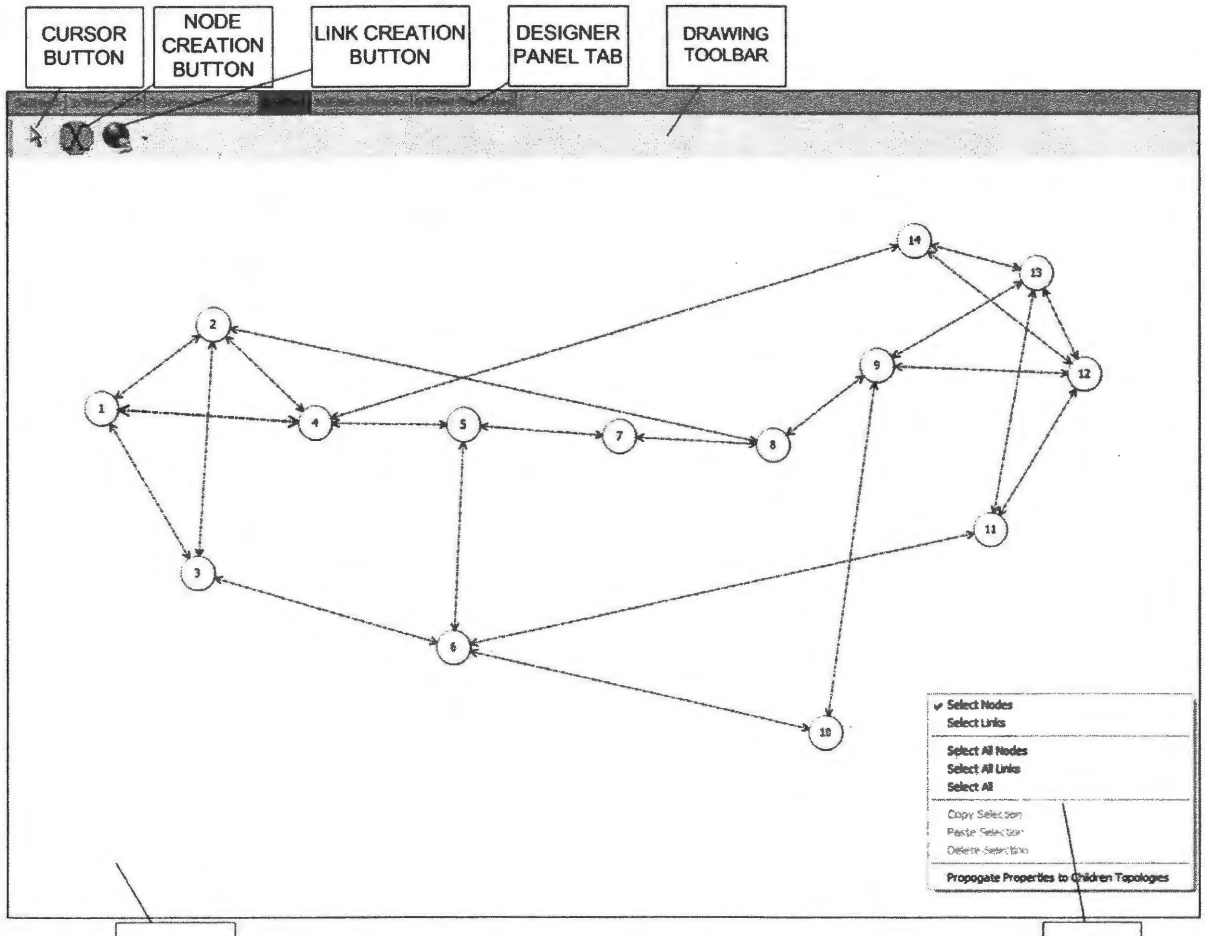


Figure 5.2. Designer panel window.

or network's properties in pop-up property box, save an experiment, and lock or unlock an experiment. Modifications to the network structures of all networks in an experiment can only be done in unlock state on the default network topology panel. When an experiment is locked, the network structure is propagated to all networks to reflect any changes made to the network topology. The default network topology is used to draw the common network structure in an experiment.

- Properties explorer

  The simulation parameters – which include the properties of the experiment, its networks' properties, and those belonging to the network components, i.e. nodes and links – can be easily set and viewed from the properties explorer window. All possible

inputs for each parameter are provided as choices in combo box. This facilitates the user by not requiring them to remember all possible simulation parameters and knowing the simulation files format.

A brief description of each object's property is displayed at the bottom of the properties explorer window to assist users in better understanding of what the property represents. A screenshot of the property explorer window is presented in Figure 5.4.

- Connection requests window

  During simulation pause, connections existing on each network at that simulation point may be viewed from the connection requests explorer window. As illustrated in Figure 5.5, these connections are listed in a tree form, just like the experiment tree in experiment explorer window. Each connection forms a new tree, and the connection's properties, i.e. the channel capacity, the route it took, the arrival and holding time, etc., are each listed as a child tree node.

  Path taken by a connection request is drawn when the request node is selected. Load on each link on every network topology is also calculated and displayed in the properties explorer window, and the relative load percentage is shown through different coloring of the links on the network's designer panel.

  Selectively chosen connections can also be found by filtering the list of accepted connection requests. Filtering can be done so that only the connections that pass through the nodes and links of user's interest are displayed on the connection requests window.

- SRLG explorer

  A set of links correlated in terms of its risk of failures may also be set as a Shared Risk Links Group (SRLG), and the SRLG window lists each of these groups and its link members in a tree format. The assignment of the SRLG groups can only be done from the experiment property pop-up window opened by right-clicking an experiment node on the experiment explorer window. Figure 5.6 shows a screenshot of the SRLG window.

- Play control box

  User starts simulation by toggling the start button on the play control box or by selecting the "Run Simulation" menu item under the "Tools" menu bar on ISTOS main menu. User may halt the simulation by clicking the pause button at any point during the simulation. Automatic pause may also be set through breakpoints. Progress bar on play control box also shows the simulation progress in 10% increment. Figure 5.7 displays a screenshot of the play control box and the breakpoint form that pops up when breakpoint button on play control is pressed.

Figure 5.8 shows a screenshot of a sample experiment window. The designer panel, the experiment explorer, the properties window, and the play control box displayed are in the same positions as at the start of the application. Any of the windows, except the designer panel, can be moved, minimized, or closed. Any closed window can be displayed again by checking the corresponding window list on "View" main menu item. Note that however, the experiment explorer window is the only place where changes to an experiment structure, i.e. addition or removal of networks within the experiment, can be done.

The large white box on the center of the application window is the designer panel. Note that the drawing toolbar displayed on top of the designer panel can be found only on the default network topology's designer panel.

The properties explorer window can be found at the bottom right corner in Figure 5.8. Any property value that cannot be modified is displayed in gray font, instead of the regular

black font. As mentioned before, the description accompanying each property field is displayed at the bottom of the properties window.

The play control box found below the designer panel contains the play/pause button, the stop button to terminate the simulation early, breakpoints button on which automated simulation breakpoints may be set, and the simulation progress box.

Figure 5.9 shows the experiment property form that is displayed when user chooses "Properties" from an experiment node's right-clicked menu in experiment explorer. The grouping of links to become an SRLG group can be done only on the SRLG box inside this form. Note that the SRLG lists can be accessed only from the default network topology tab. Because the structures of the networks within an experiment have to remain the same, the SRLG groups must then belong to the experiment rather than the network.

The connection requests window can be seen in Figure 5.10. Each tree node contains the detailed properties of a connection in progress. A + sign on the tree node means that the node can be expanded. By clicking the Filter button at the top of the window, user can specify the nodes and/or links of interest to them, and this is useful for efficient search within the typically long list of connections.

The path taken by connection R447 is traced on the dash line when any property node under the connection node "R447" is highlighted. All links in the network are colored according to their relative loads.

Output window

The output window presents the simulation results in table format. Each metric is presented in one column and the simulation result for each round is displayed in one row. At the last row, mean value of each performance metric is calculated. User can choose which round results to be included in the average calculation. Output file corresponding to each

network topology is placed on a sub-window on the output window and is accessible through its tab. An output window screenshot is shown in Figure 5.11.

5.1.2 Experiment

An *experiment* is the base unit for simulation in ISTOS. User can create more than one experiment on one ISTOS application run, but only one experiment can be simulated at any time. An ISTOS experiment contains a default network topology and one or more network topologies that need to be compared. Each network topology has the same network structure as the default network topology but different network properties and parameters. The multiple network topologies comprising an experiment are simulated simultaneously. This allows performance evaluation of different properties in one simulation run to determine which configurations best suit a network topology.

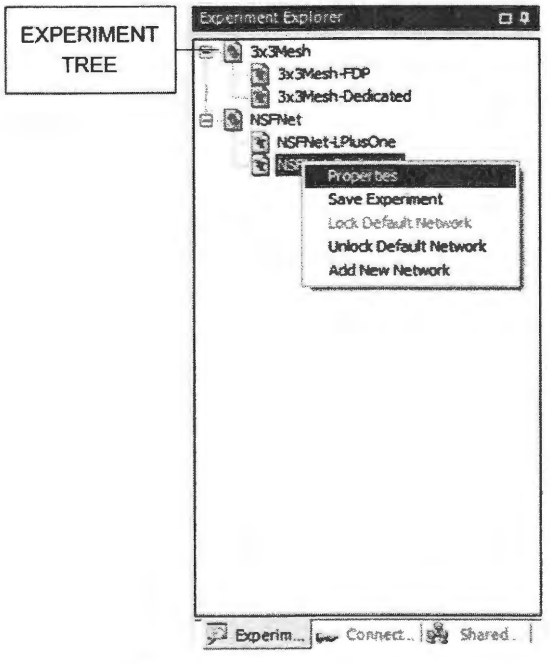


Figure 5.3. Experiment explorer window.

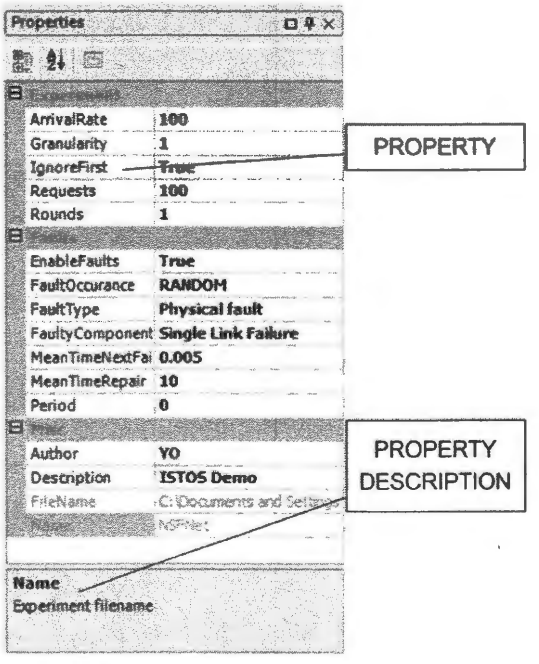


Figure 5.4. Properties explorer window.

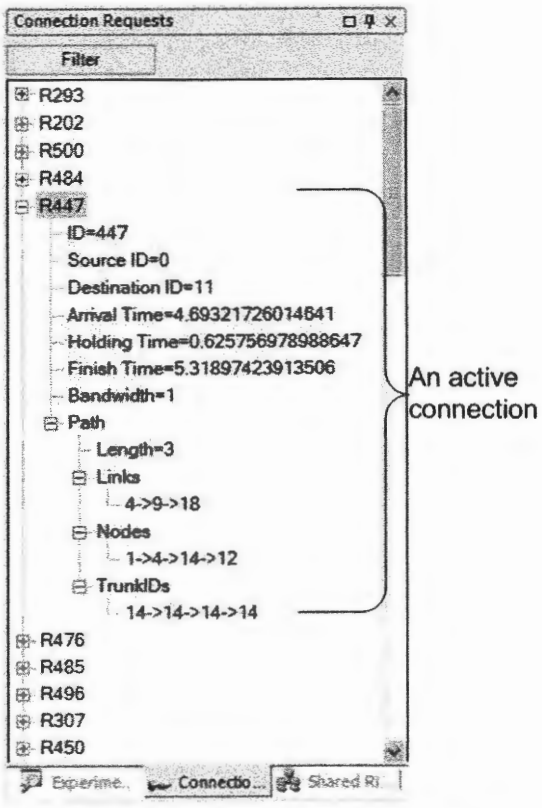


Figure 5.5. Connection requests window.

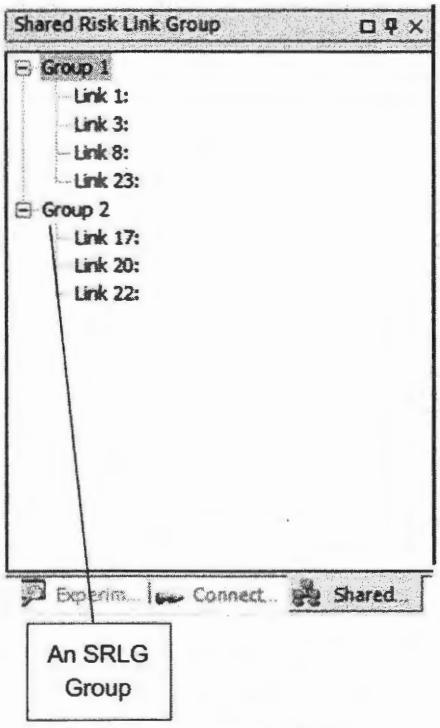


Figure 5.6. SRLG window.

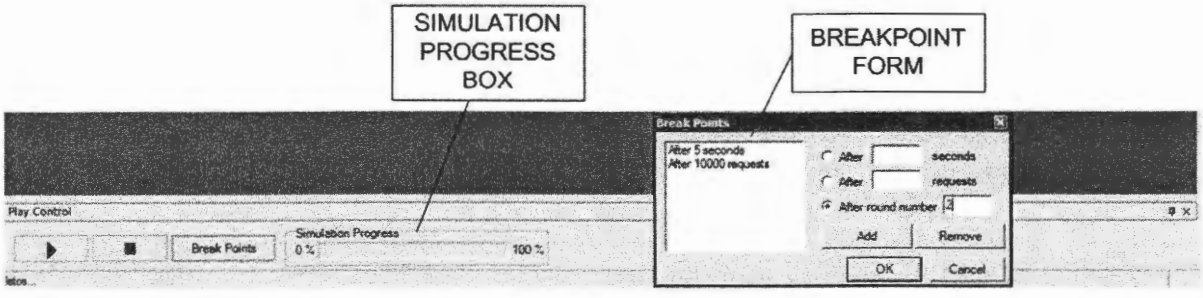


Figure 5.7. Play control box.

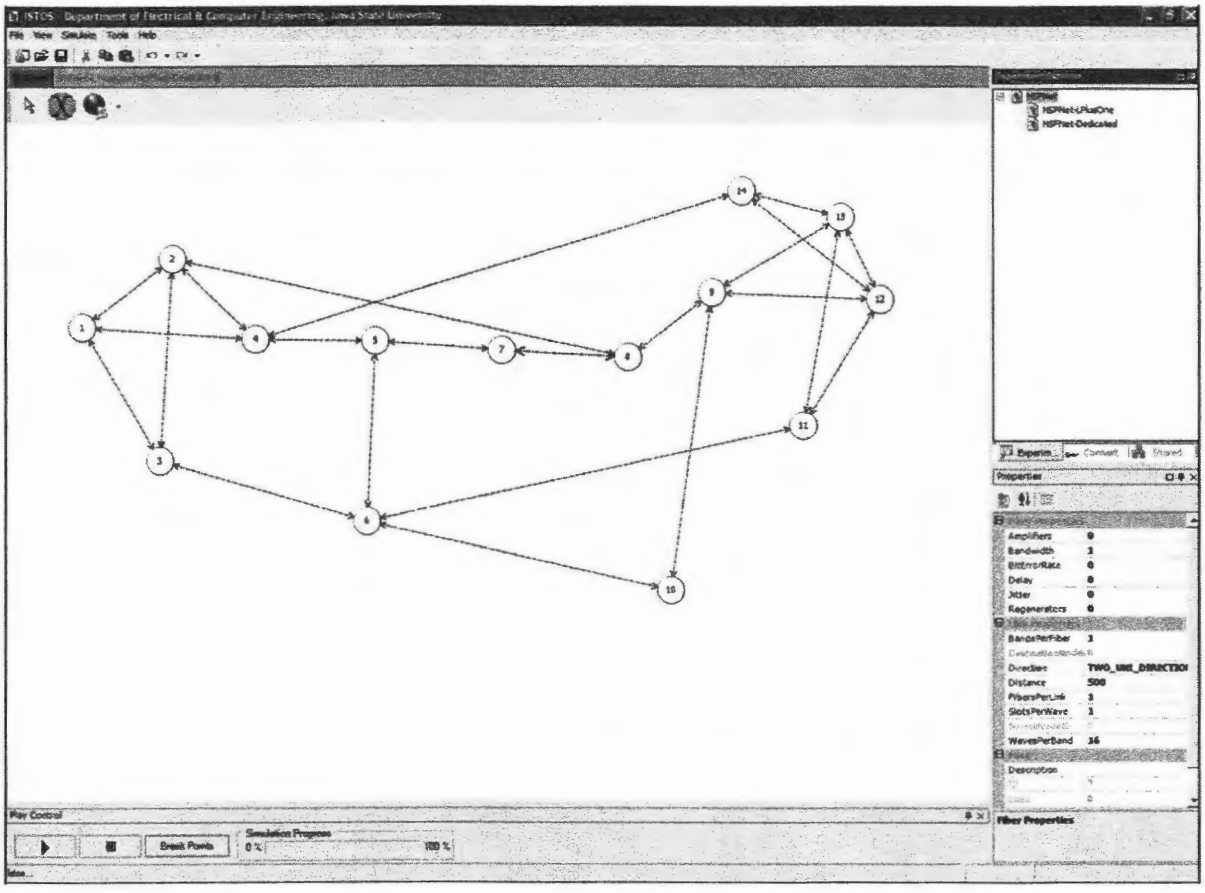


Figure 5.8. Screenshot of the GUI window of an example experiment.

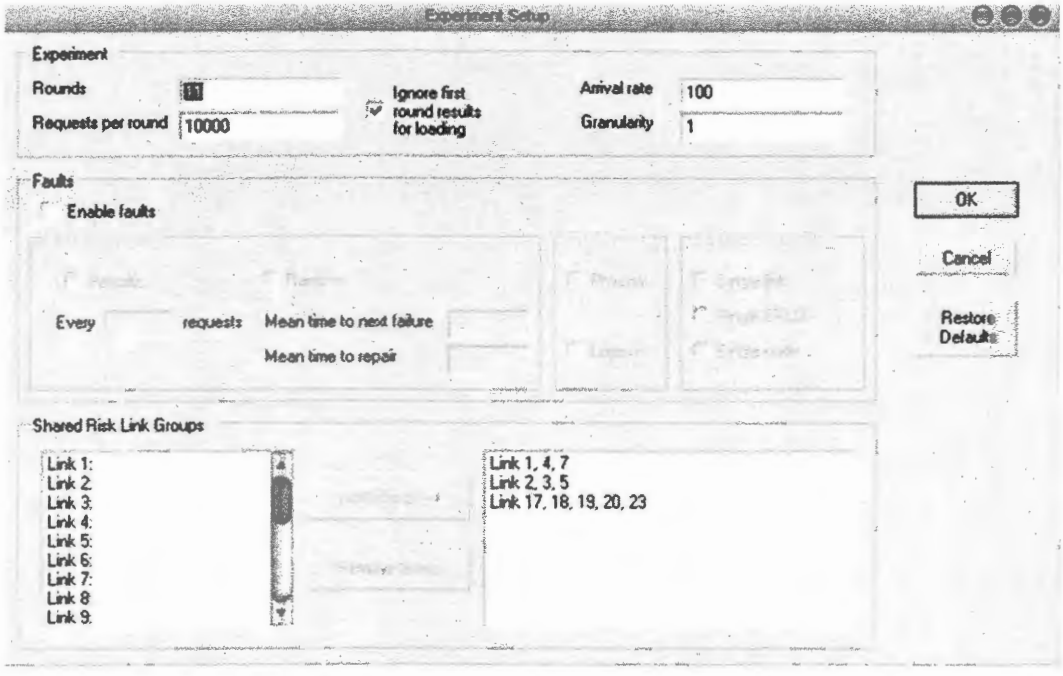


Figure 5.9. Experiment pop-up property form.

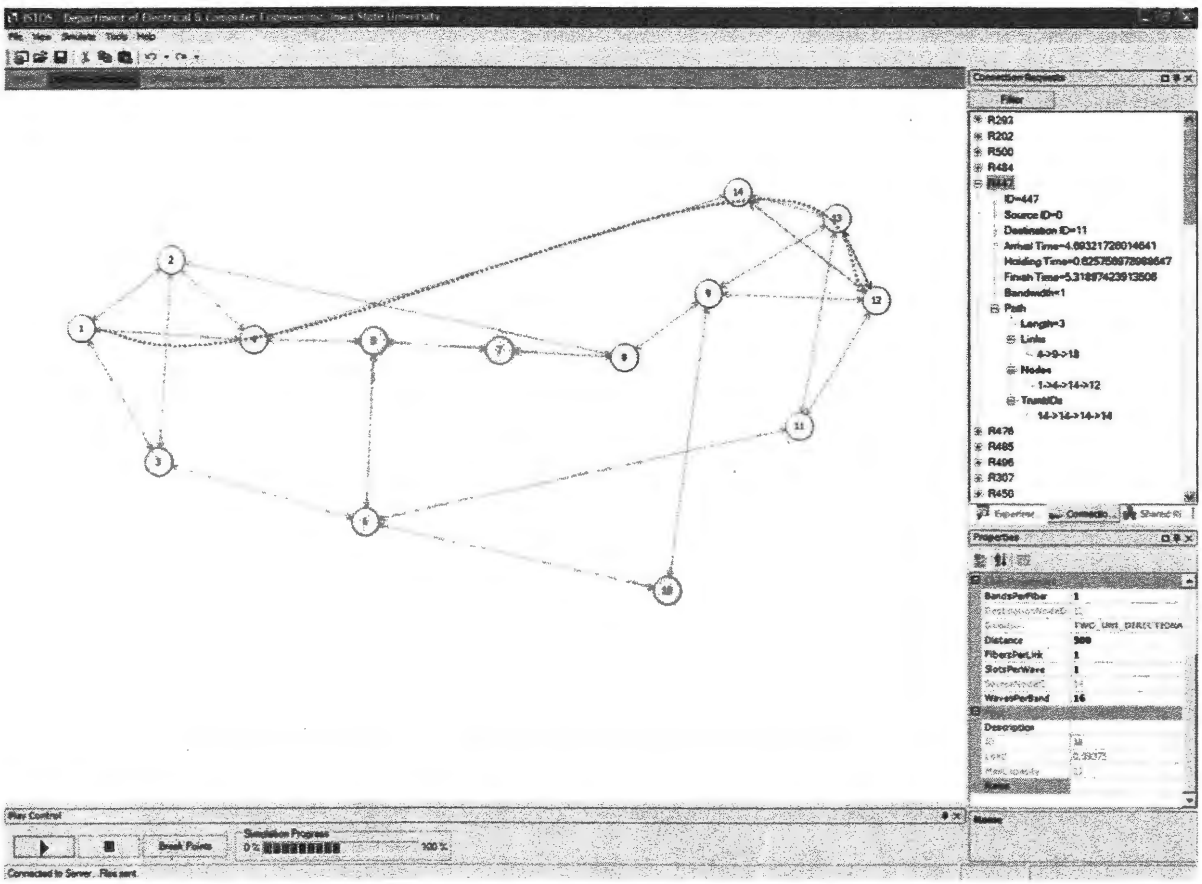
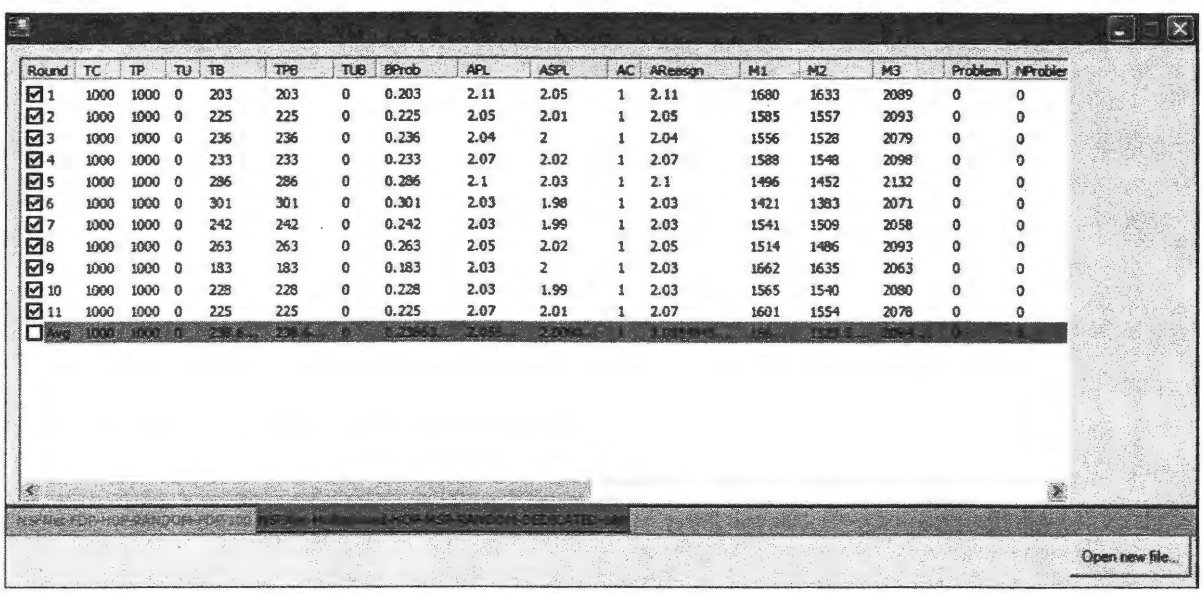


Figure 5.10. Network status showing relative link load and connection requests present during simulation pause.



Round	TC	TP	TU	TB	TPB	TUB	BProb	APL	ASPL	AC	AReassgn	M1	M2	M3	Problem	NProbler
1	1000	1000	0	203	203	0	0.203	2.11	2.05	1	2.11	1680	1633	2089	0	0
2	1000	1000	0	225	225	0	0.225	2.05	2.01	1	2.05	1585	1557	2093	0	0
3	1000	1000	0	236	236	0	0.236	2.04	2	1	2.04	1556	1528	2079	0	0
4	1000	1000	0	233	233	0	0.233	2.07	2.02	1	2.07	1588	1548	2098	0	0
5	1000	1000	0	286	286	0	0.286	2.1	2.03	1	2.1	1496	1452	2132	0	0
6	1000	1000	0	301	301	0	0.301	2.03	1.98	1	2.03	1421	1383	2071	0	0
7	1000	1000	0	242	242	0	0.242	2.03	1.99	1	2.03	1541	1509	2058	0	0
8	1000	1000	0	263	263	0	0.263	2.05	2.02	1	2.05	1514	1486	2093	0	0
9	1000	1000	0	183	183	0	0.183	2.03	2	1	2.03	1662	1635	2063	0	0
10	1000	1000	0	228	228	0	0.228	2.03	1.99	1	2.03	1565	1540	2080	0	0
11	1000	1000	0	225	225	0	0.225	2.07	2.01	1	2.07	1601	1554	2076	0	0
Avg	1000	1000	0	236.4	236.4	0	0.2364	2.065	2.0091	1	2.065	1601	1554	2076	0	0

Open new file...

Figure 5.11. ISTOS output window.

5.2 Back-End Core Simulator

The back-end simulation engine contains three main modules: the event generator, the algorithms library, and the simulation monitoring system.

5.2.1 Event Generator

The event generator comprises of the request and fault generator modules. Either a request or a failure is generated as an event and passed to the algorithms library to be handled. Therefore, the request generator and fault generation module have to collaborate to fabricate only one event at a time.

Request Generator

The request generator generates connection requests to the simulator core. Traffics generated are based on the following assumptions:

1. Poisson arrival rate

   Connection requests arrive at network according to Poisson process with rate λ as specified by front-end user.

2. Exponential service time

   The holding time of requests are exponentially distributed with mean of 1.0.

   The load on the network depends on both λ and μ, as defined in Equation (9) in Section 4.4, but since μ is a constant, the arrival rate parameter can be varied to control the load on the network.

3. Source and destination nodes for a request are generated using uniform probability distribution, i.e. a request can go from any node to any other node with equal probability.

4. Request bandwidth is also generated using uniform probability distribution that ranges from 1 to the maximum bandwidth granularity specified by user, which we refer to as *request granularity*.

Fault Generator

The fault generator is used to inject faults into networks if failure scenarios are enabled by users.

There are three distinct fault components that can be simulated:

1. Node

   When a node fails, all links connected to it are simulated to experience failures. Hence, a node failure corresponds to multiple link failures with node as a common failure structure.

2. Link

   Only a single link failure can be simulated if a link component is said to be failed.

3. Shared Risk Link Group (SRLG)

   One of the SRLG groups pre-defined by user is selected to be failed. This, again, is equivalent to simulating multiple link failures with the exception that these links have to have a common failure-vulnerable structure.

User may specify any combination of the above fault components to be the components that are subjected to failure in an experiment.

The two types of failures that can be simulated in ISTOS are physical and logical failures. If logical failure is chosen, the logical topology of the network structure is used to determine

the failure probability of the components that are susceptible to failure. In other words, a two unidirectional link is said to have twice the probability of being failed than a unidirectional link, a bidirectional link, a node, or an SRLG group.

There are two types of failure duration that can be simulated in ISTOS:

1. Periodic

   In periodic failure, fault is simulated on each network after every multiple of user-specified number of requests (referred as *fault period*) are generated.

2. Random

   If random failure is used, a component to be failed is chosen in random. All pre-determined failure-susceptible components have equal probability of being failed.

   Failure arrival rate and holding time follows the exponential distribution with user-specified mean time between failures and mean repair time variables. There is at most one failure event presents in an experiment at any point in time during simulation. If a second random failure is generated to arrive before the previous one terminates, the failure is ignored and the next failure is regenerated.

After a failure event is simulated, the network reconfiguration module is run to evaluate the effect of the failure on existing connections.

5.2.3 Algorithms Library

The algorithms library contains the routing module, the trunk assignment module, the network survivability design, and the network reconfiguration module. On every request arrival, the routing and trunk assignment modules are run to obtain the path and subtrunks assigned to connect the request. Some routing module, such as shortest path routing, routes connection regardless of the network state. Other routing modules take network state into

consideration in finding "the most suitable" route from a set of potential paths. For example, widest shortest path routing returns the path with the widest path information vector; this path selection scheme is used to increase the possibility of finding enough resources, i.e. link channels in this case, to route the connection. Path information matrices computed on the MICRON framework are incorporated as inputs to this scheme. The routing and trunk assignment protocols to be used on a network simulation are specified by the user as network's properties.

The network survivability design module contains the network protection algorithms that encompass fault tolerance in designed networks. This module might interact with the routing and trunk assignment modules in order to set up backup paths that are usually needed to guarantee some degree of protection against failures in the network.

Network reconfiguration module is run for every failure generated to check if any existing connections on the network are disrupted because of the failure. The compromised connections, if any, are removed from the network and the quantity of these connections are calculated and send to the simulation monitoring system module.

5.2.4 Simulation Monitoring System

The connections list holder module keeps track of the connections existing on the network at a particular point in simulation. This list is sent to the front end everytime the simulation is paused (manually or by breakpoints). This module also interacts with the network reconfiguration module as it needs to remove the connections disrupted in response to network failures.

The metric processing module computes statistical result data based on each connection request's status information (whether a connection is accepted or rejected) and properties, which are fed from the algorithms library module.

5.3 IstosComm

IstosComm is a TCP-based customized socket that defines the communication protocol between the front end and the back end of ISTOS. When user decides to run the simulation, the front-end sends the network structure information as well as the experiment and the network simulation properties as configuration files to the backend. During pause and end of simulation, the backend sends the on-line network status and output file(s), respectively, to the front end. The IstosComm also allows the transmission of commands that makes possible the ability to halt simulation and terminate simulation run at any point per user's desire.

5.4 Summary

In this chapter, we described the front-end user interface and the back-end modules that shape ISTOS, and the customized TCP-based protocol that provides flexibility to users to halt simulation, get on-line network status, and terminate the connections prior to end of simulation.

We explained the three main modules of the backend: the event generator, the algorithms library, and simulation monitoring system. These modules are responsible for generating requests traffic and failures, handling the routing and trunk assignment for primary path and backup paths, if required, and reconfiguring the network during failures, and maintaining the list of active connections as well as performance metrics for each network under evaluation, respectively.

We also showed screenshots of windows and forms that user can access from the front-end GUI that assists in providing user-friendly environment in ISTOS.

CHAPTER 6 Simulation Examples

ISTOS provides a user-friendly design environment by providing menu bar and toolbars which are very intuitive to use Figure 6.1 illustrates a 15-node, 19-link general network topology named GEANTNet on ISTOS. GEANTNet is a part of the GÉANT network, a pan-European multi-gigabit data communications network specially reserved for research and education purposes, which is connected by the 10 Gbps data link [6].

To start a new experiment like this, user has to click on the "New Experiment" button or its corresponding menu item under "File" menu bar. A pop-up window, as shown in Figure 6.2, would then appear, asking if a blank experiment or one based on a template should be created. A *template* provides the network topology structure for an experiment. This is significantly useful when simulating networks with similar but not identical topology. Instead of redrawing the whole topology, the topology structure based on the template is edited to create the new network topology.

The creation and modification of template is also possible on ISTOS GUI. Any template opened in ISTOS, however, is not recognized as a new node in the experiment tree on experiment explorer window. Therefore, saving and closing of a template can only be done from their respective menu items under "File" menu bar. Any opened experiment can also be saved as an ISTOS template file with extension .ITP.

A form as shown in Figure 6.3 would pop up after user chooses either of the new experiment options in Figure 6.2. This is the basic experiment property box in which user specifies the name of the experiment, the location to save all its files, and extra details such as name of the author and description or side notes on the experiment. The name and the location fields are not modifiable once they are set.

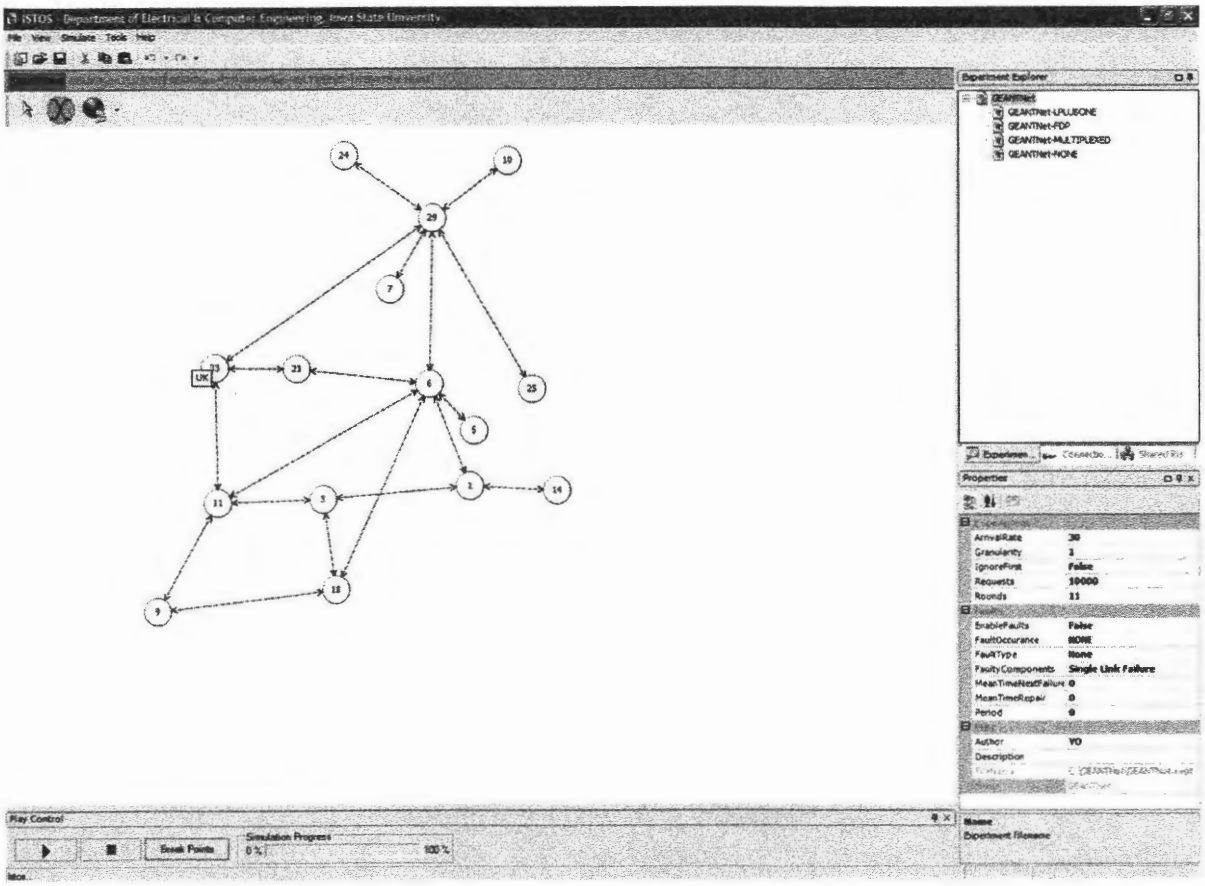


Figure 6.1. 15-node, 19-link GEANTNet.

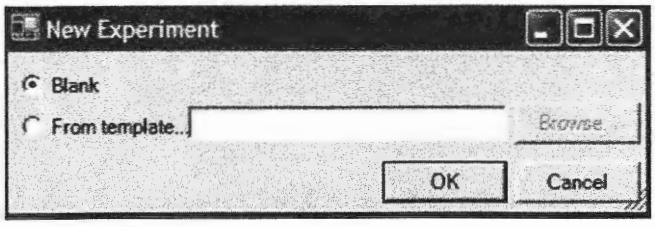


Figure 6.2. New experiment form.

The default network topology window is the first window user shall see when an experiment is opened. Drawing of nodes and links that make up the network structure can then be done on its designer panel.

The experiment shown in Figure 6.1 consists of four network topologies named GEANTNet-LPLUSONE, GEANTNet-FDP, GEANTNet-MULTIPLEXED, and GEANTNet-NONE. Any experiments and network topologies currently opened are shown on experiment explorer window in their hierarchical order. Only on this window, user may

lock or unlock the default network topology (shown as GEANTNet in this case) to add or remove a new or existing network topology.

The GEANTNet experiment's property is shown on properties window in Figure 6.1. Other objects, such as node, link, or network's properties that determine the simulation parameters are also listed here. The network or experiment's properties can also be viewed on a pop-up property window opened from the pop-up menu in experiment explorer window. The pop-up menu can be accessed by right-clicking the node in the experiment tree. A node or link's properties are also available on the pop-up property window which appears when user double clicks on the node or link. This property window is not accessible for the nodes and links in a locked default network topology since their properties are not modifiable there.

Besides its unique id, a node or link can be identified by its name. If a node is named, its name would appear whenever cursor points at the node, just as shown in Figure 6.1. This feature is especially helpful to associate the network drawn on ISTOS with the real network topology to be copied.

Complete step by step procedures on how to run simulations in ISTOS and what features that ISTOS supports are provided in ISTOS help files accessible from the "Help" menu bar. Any ISTOS-related terms and definitions are also described in these files. Fast and easy discovery of a topic is provided through two different ways of presenting the help files: by contents and by topic index. The description of each simulation parameter or object's property available at the bottom of the properties window is also helpful in avoiding unnecessary misconception of terms definition. The ISTOS help window is shown in Figure 6.4.

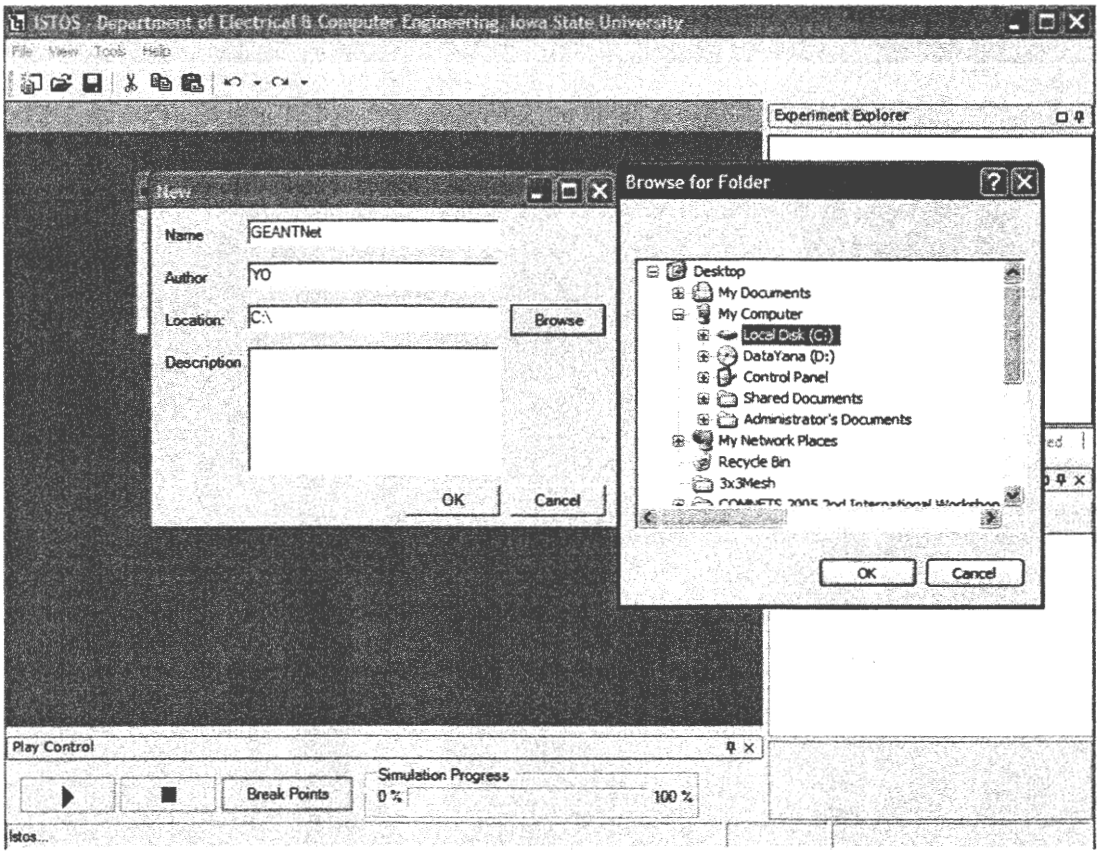


Figure 6.3. New experiment property form.

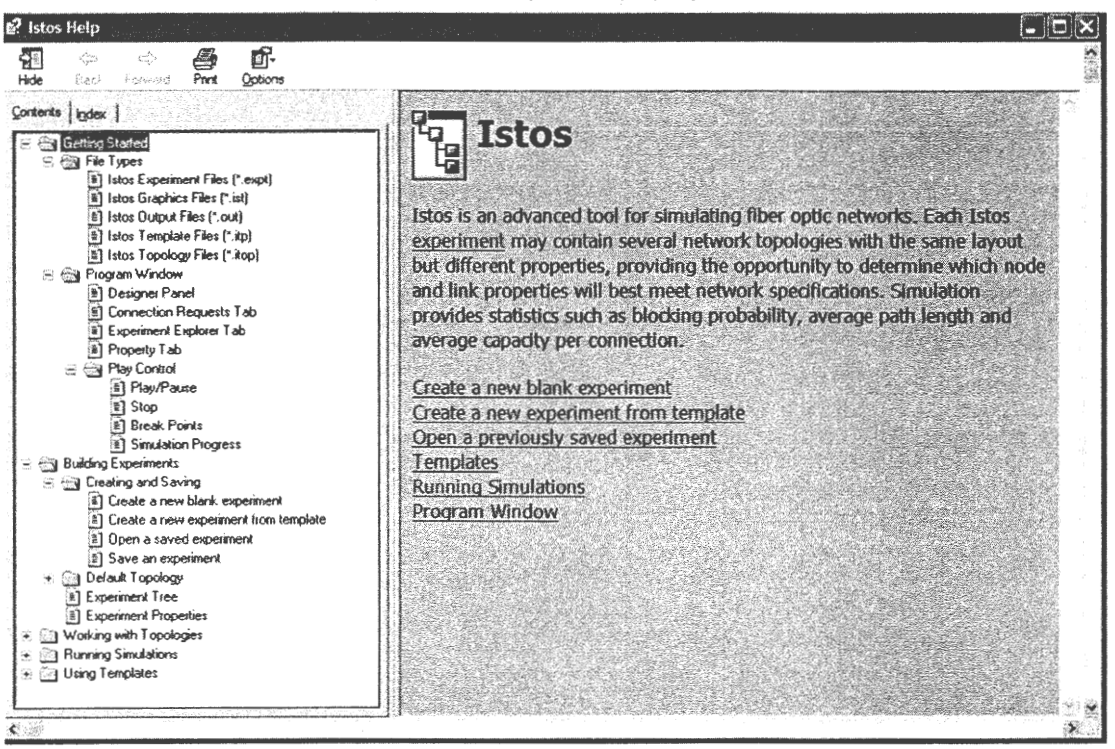


Figure 6.4. Help window searchable by contents or index.

6.1. Performance Evaluation

We run experiments on three different networks: NSFNet, 3x3 mesh torus and GEANTNet network, to illustrate the type of performance evaluations that can be done by using ISTOS. The topologies of the networks are shown in Figure 6.5, 6.6 and 6.1, respectively. Each network is simulated as homogeneous network with fiber and time-slot conversion capabilities on each node. Every link in the network employs two-unidirectional fibers, each consisting sixteen wavelengths and one time-slot per wavelength.

All networks are simulated employing different path protection strategies against single link failure. We simulate dedicated backup, backup multiplexing, and failure-dependent path protection (FDP) strategies in 3x3 mesh network. For NSFNet and GEANTNet networks, we run backup multiplexing, FDP, and (L+1) subgraph routing strategies. We also simulate networks without any backup strategy for comparison purposes in all our experiments.

The NSFNet and GEANTNet networks are simulated using the available shortest path (ASP) routing strategy with random trunk assignment for the primary and backup, if applicable, connections. 3x3 mesh torus, on the other hand, is simulated employing shortest path on hop count with random trunk assignment.

Each experiment is run for 11 rounds; the analysis discussed in the following does not include the first round computational results. The request granularity for all the experiments is set to 1. Experiment is simulated at various traffic arrival rates, depending upon connectivity of the network, with holding time of 1.0. Network with high level of connectivity such as 3x3 mesh torus is simulated with higher traffic arrival rate since a connection request has higher chance of being routed due to higher number of potential paths in the network.

Figure 6.7 shows the blocking probability in NSFNet network. As shown in Figure 6.7, the FDP and (L+1) subgraph protection strategies perform reasonably well compared to backup multiplexing strategy. One factor to this observation is the relatively low level of

connectivity in NSFNet network in which it does not guarantee there are at least two link-disjoint paths between any two nodes.

Figure 6.8 shows the blocking probability for 3x3 mesh torus network. The blocking probability for dedicated backup and backup multiplexing in 3x3 mesh torus are relatively similar, while the FDP has the highest blocking probability in this case. Both dedicated backup and backup multiplexing performs very well in comparison to FDP in this case since each node in 3x3 mesh torus has exactly four link-disjoint paths to every other nodes. Thus, for each primary connection established for a request, it has a higher chance of establishing a link-disjoint backup path than establishing multiple not-necessarily link-disjoint backup paths.

Figure 6.9 shows the blocking probability for GEANTNet network. In this case, the performance result for backup multiplexing, FDP, and (L+1) protection strategies appear to be very similar and they are performing relatively poor. One factor to this observation is the network has a very low connectivity level. It also appears that the network has reached the saturation point in terms of the number of requests that it can accept for entire arrival rate interval.

Figure 6.10, 6.11, and 6.12 show the effective utilization for NSFNet, 3x3 mesh torus, and GEANTNet network, respectively. The effective utilization metric is normalized with respect to the simulation duration and total link capacity in the network. Since the number of requests per round is set to be constant for all experiments, we expect to see increasing effective utilization, as observed, as the traffic arrival rate increases due to the shorter simulation duration. As confirmed earlier, both the dedicated backup and backup multiplexing outperform the FDP strategy in 3x3 mesh torus network, while the (L+1) subgraph strategy outperforms both FDP and backup multiplexing in NSFNet. As seen in Figure 6.12, all backup strategies perform very poorly in GEANTNet and less than 10% of the network resource is utilized for all arrival rates.

Data for the simulations based on Figure 6.7 to Figure 6.12 are drawn can be seen in Appendix.

Table 1 shows the average path length for all accepted connections in NSFNet and Table 2 shows its average shortest path length. The average path length increases as arrival rate increases. This might be due to the way available shortest path (ASP) algorithm works. Since ASP routing algorithm routes connection on the shortest path out of all available paths, as arrival rate increases, connections have to be routed on a longer path around existing connections.

The average path length in 3x3 mesh torus is shown in Table 3. Since shortest path on hop count is used for routing both primary and backup connections, this table also represents the average shortest path length. The average path length decreases as arrival rate increases. This might be because the number of requests accepted in the network decreases and consequently, the number of connections for a request to have to route around is lower.

The average path length and average shortest path length of all accepted connection requests in GEANTNet are shown in Table 4 and Table 5, respectively. We observe the same results as in NSFNet network in this case since they both employ the same routing algorithm.

Table 6 shows the average number of path reassignments for (L+1) subgraph and FDP protection strategies in NSFNet network. FDP strategy generates backup paths only for each possible failure scenario for a request, and thus the number of path reassignments is at most the number of links the primary path is routed on. On the other hand, the (L+1) subgraph strategy routes each request on the subgraphs independently from the path taken by the request in base network. The number of path reassignments is thus very high. We observe the same results in GEANTNet network, as shown in Table 7.

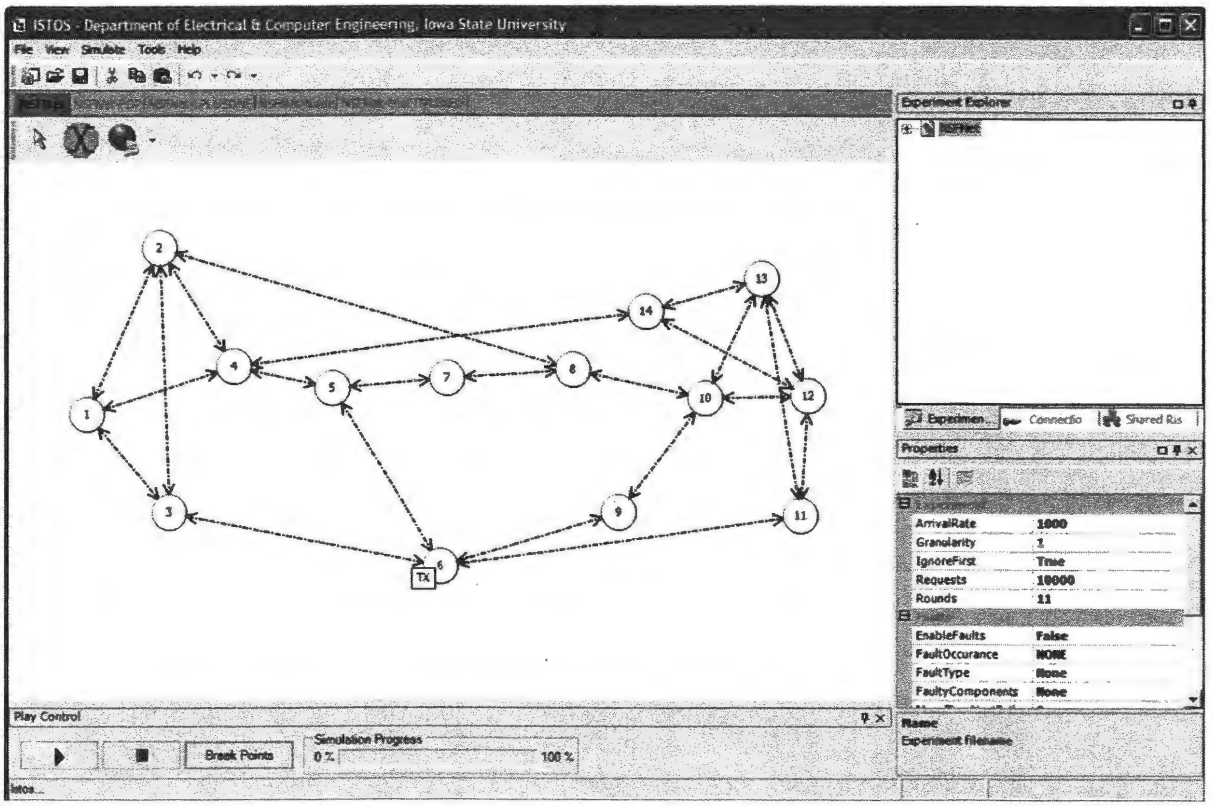


Figure 6.5. 14-node, 23-link NSFNet.

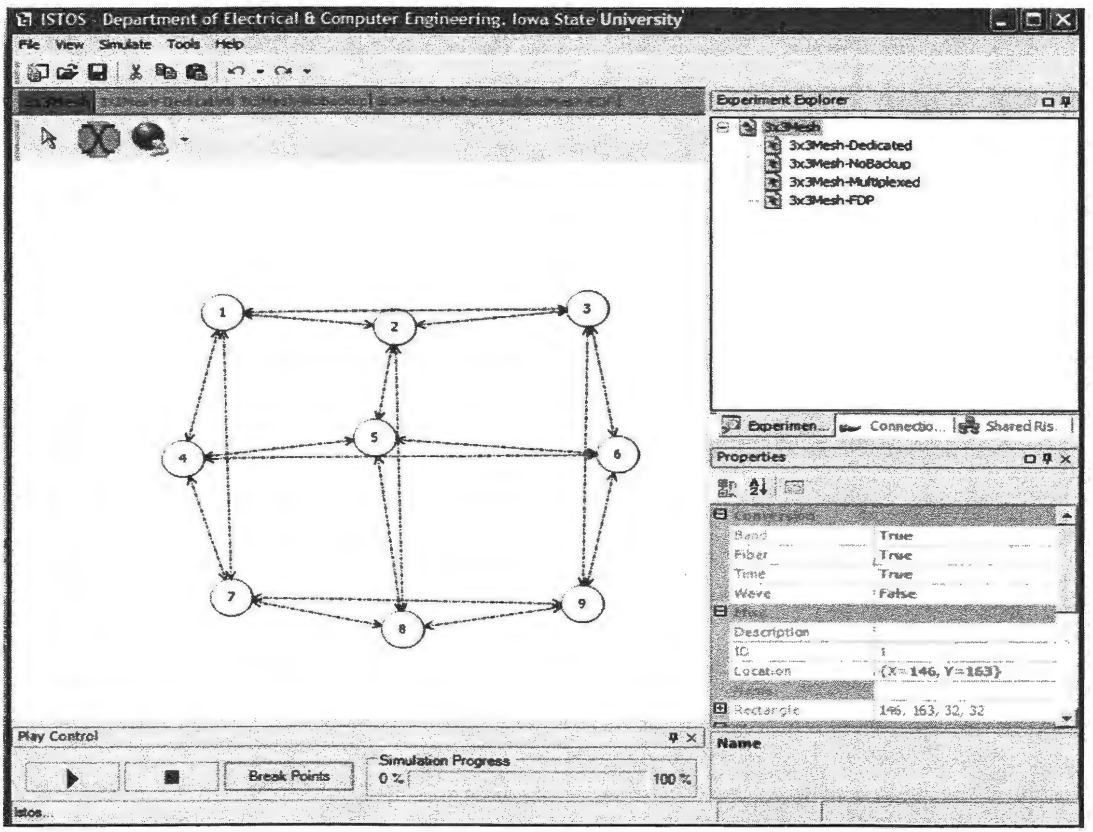


Figure 6.6. 9-node, 18-link 3x3 mesh torus.

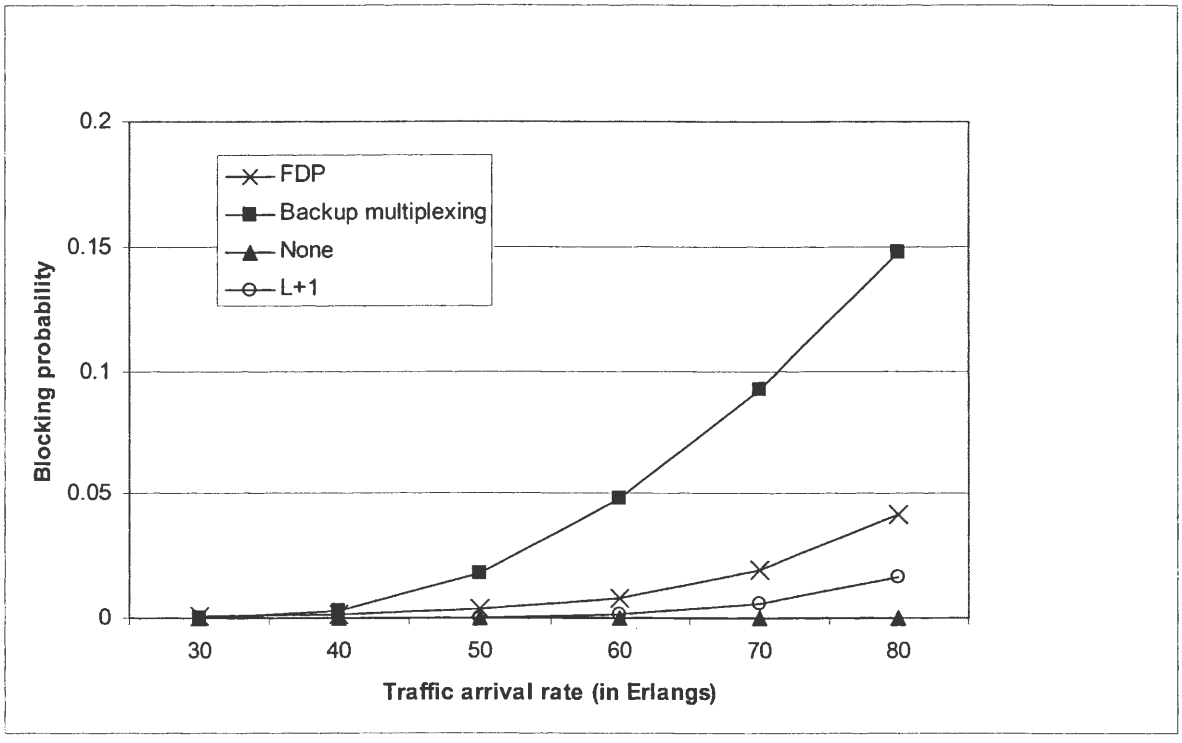


Figure 6.7. Blocking probability vs. traffic arrival rate for NSFNet.

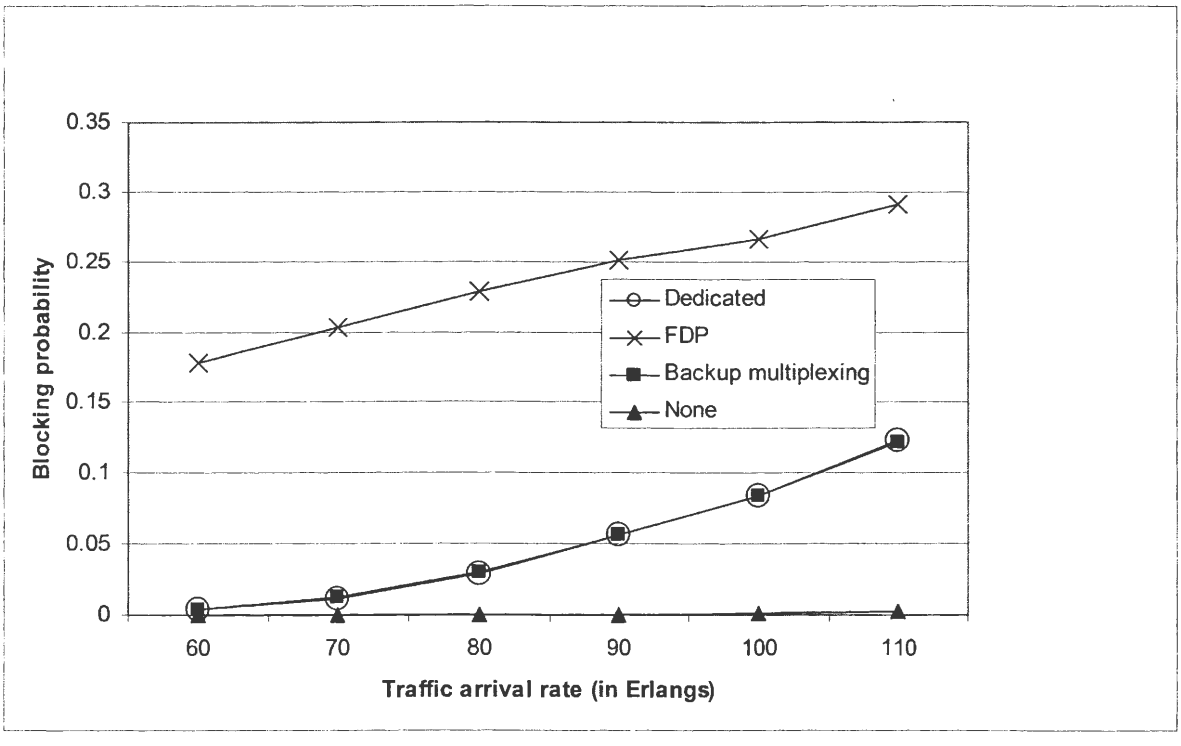


Figure 6.8. Blocking probability vs. traffic arrival rate in 3x3 mesh torus.

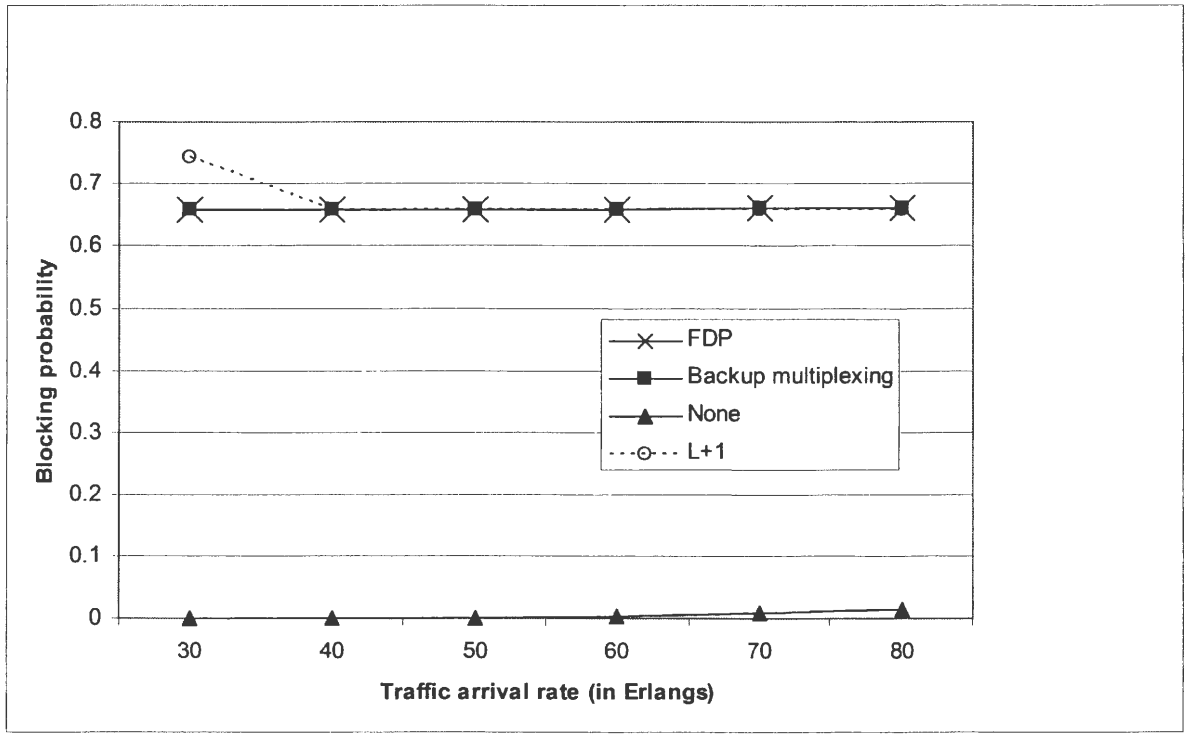


Figure 6.9. Blocking probability vs. traffic arrival rate for GEANTNet.

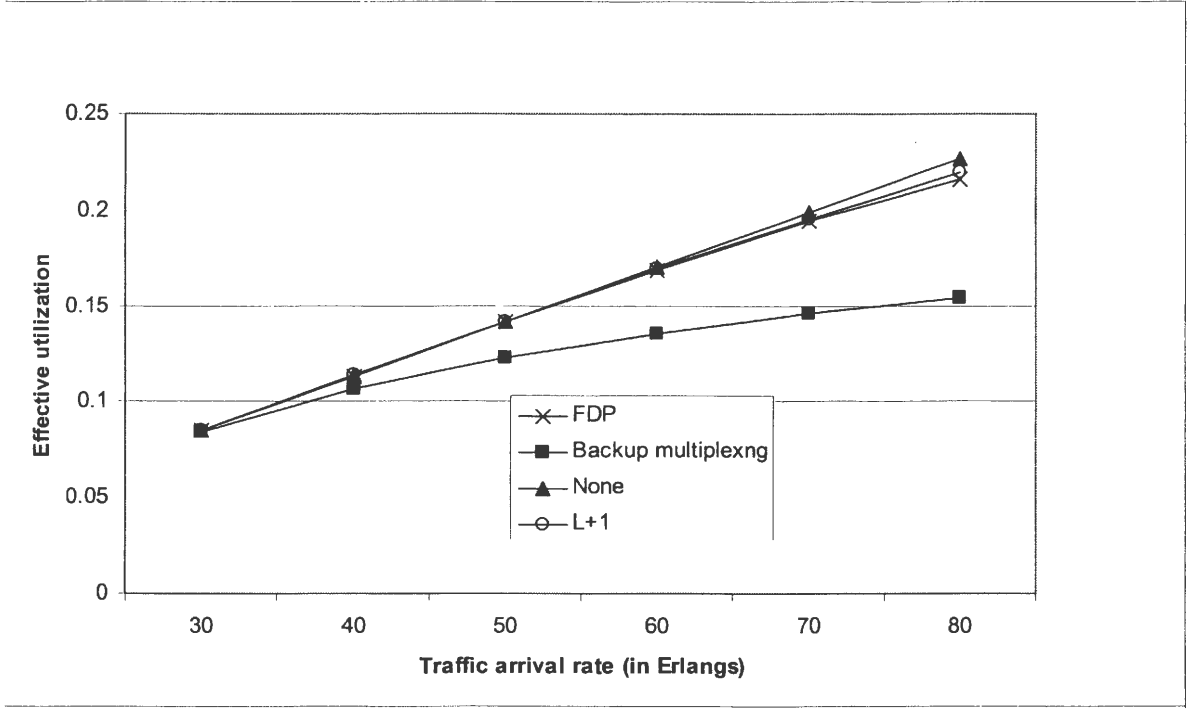


Figure 6.10. Effective utilization vs. traffic arrival rate for NSFNet.

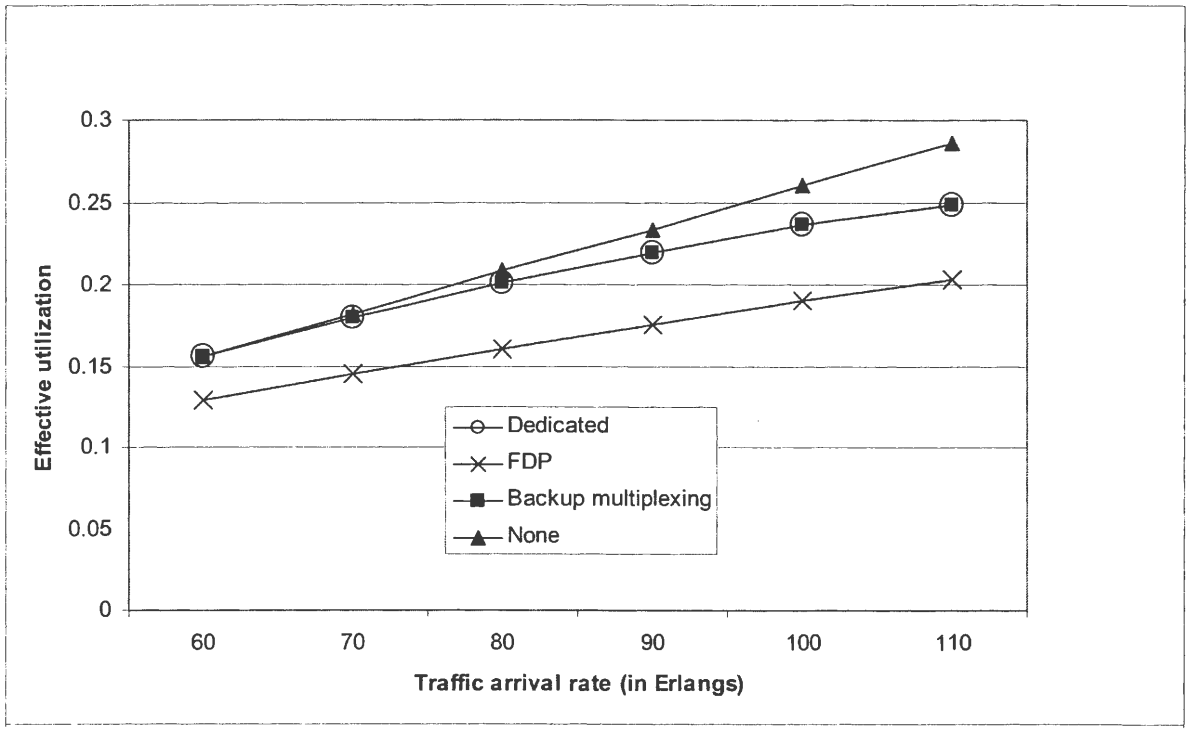


Figure 6.11. Effective utilization vs. traffic arrival rate for 3x3 mesh torus.

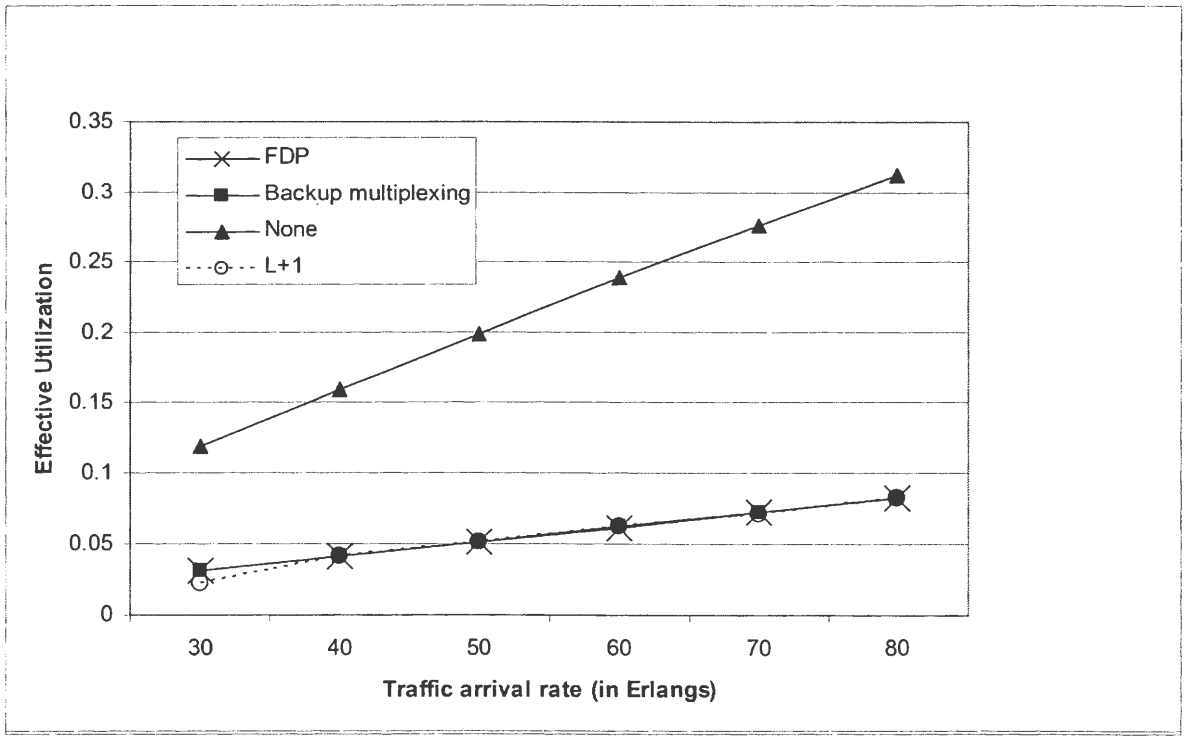


Figure 6.12. Effective utilization vs. traffic arrival rate for GEANTNet network.

Table 1. Average path length of accepted requests in NSFNet network.

Arrival Rate	30	40	50	60	70	80
None	2.089	2.089	2.086	2.088	2.09	2.09
FDP	2.089	2.089	2.094	2.105	2.125	2.145
Backup multiplexing	2.089	2.089	2.091	2.095	2.098	2.092
L+1	2.089	2.089	2.095	2.105	2.129	2.15

Table 2. Average shortest path length of accepted requests in NSFNet.

Arrival Rate	30	40	50	60	70	80
None	2.089	2.089	2.086	2.088	2.088	2.089
FDP	2.089	2.089	2.085	2.086	2.082	2.071
Backup multiplexing	2.089	2.089	2.085	2.081	2.07	2.056
L+1	2.089	2.089	2.085	2.086	2.081	2.069

Table 3. Average path length of accepted requests in 3x3 mesh torus.

Arrival Rate	60	70	80	90	100	110
None	1.499	1.499	1.498	1.498	1.499	1.498
Dedicated	1.498	1.498	1.494	1.488	1.489	1.48
Backup multiplexing	1.499	1.498	1.495	1.488	1.488	1.48
FDP	1.503	1.499	1.499	1.496	1.498	1.498

Table 4. Average path length of accepted requests in GEANTNet network.

Arrival Rate	30	40	50	60	70	80
None	2.421	2.414	2.423	2.433	2.441	2.451
FDP	1.829	1.822	1.838	1.836	1.842	1.837
Backup multiplexing	1.829	1.822	1.836	1.834	1.843	1.835
L+1	1.732	1.829	1.83	1.835	1.821	1.831

Table 5. Average shortest path length of accepted requests in GEANTNet network.

Arrival Rate	30	40	50	60	70	80
None	2.089	2.089	2.086	2.088	2.088	2.089
FDP	2.089	2.089	2.085	2.086	2.082	2.071
Backup multiplexing	2.089	2.089	2.085	2.081	2.07	2.056
L+1	2.089	2.089	2.085	2.086	2.081	2.069

Table 6. Average number of path reassignments for accepted requests in NSFNet.

Arrival Rate	30	40	50	60	70	80
L+1	21.7	21.6	21.66	21.69	21.69	21.68
FDP	2.089	2.089	2.094	2.105	2.125	2.145

Table 7. Average number of path reassignments for accepted requests in GEANTNet.

Arrival Rate	30	40	50	60	70	80
L+1	17.9	17.91	17.91	17.9	17.84	17.9
FDP	1.829	1.822	1.838	1.836	1.842	1.837

6.2. Summary

In this chapter, we presented a short instruction on how to use ISTOS to model a WDM grooming network. We also presented the simulation results run on three networks: NSFNet, 3x3 mesh torus, and GEANTNet networks, and discussed briefly on the results. Through the examples shown in this chapter, we illustrate the efficiency and effectiveness of modeling and simulation run on ISTOS.

CHAPTER 7 Conclusions and Future Work

Optical wavelength division multiplexing (WDM) networks have appeared to be the promising technology for high-speed backbone networks. The improvement in optical technologies has allowed tremendous increase in bandwidth capacity transmitted on a wavelength. Meanwhile, end users' requirement for bandwidth does not rise at the same rate. The bandwidth mismatch calls for wavelength sharing methods. WDM networks employ time division multiplexing (TDM) technique to groom sub-wavelength connections onto a wavelength; these networks are referred as WDM/TDM or WDM grooming networks.

In this thesis, we present a software tool that is able to model both homogeneous and heterogeneous WDM grooming networks. This software, called ISTOS, provides a common platform for evaluating different technology options that address network design and operational issues.

ISTOS allows simulation of multiple networks with different resource distributions to solve the resource dimensioning problem. The goal here is to find the most effective resource distribution that guarantees a relatively high acceptance probability for future connection request traffic.

ISTOS also allows simulation of multiple networks with different operational algorithms to evaluate the most suitable operational policy on a particular network topology. The types of operational policies that could be evaluated in ISTOS include the routing and trunk assignment schemes as well as network protection strategies.

We recognize some possible extensions to ISTOS that can enhance the overall features that the software offers. A more user-friendly way of adding a new routing, trunk assignment or network protection algorithm can be done by providing an application user interface (API).

Currently, ISTOS only provides traffic and failure generation that follows the Poisson distribution function. Generation of request traffic and faults that is based on other random distribution pattern may be desirable to offer a wider variety of dynamic traffic and faults.

Many additional features can be integrated in ISTOS user interface to provide an even more user-friendly environment. A history of tasks done on the designer panel could be maintained to provide the ability to "undo" and "redo" on the GUI. That way, user does not have to redo the whole topology drawing if he/she accidentally deletes the nodes and links in the network.

ISTOS backend runs only on single CPU regardless of the number of networks being simulated. This might degrade the overall performance of the software. This is especially true if a network with (+1) subgraph routing is simulated. This is due to the typically large number of network states that have to be maintained for this protection strategy. An option to run simulations on different machines is to be investigated. One possible solution is to run ISTOS on a P2P-based framework developed in [14] called CompuP2P.

BIBLIOGRAPHY

[1] A. Aggarwal, A. Bar-Noy, D. Coppersmith, R. Ramaswami, B. Schieber, and M. Sudan. "Efficient routing in optical networks." *Journal of the ACM*, 46.6 (1996): 973 – 1001.

[2] A. E. Ozdaglar and D. P. Bertsekas. "Routing and wavelength assignment in optical networks." *IEEE/ACM Transactions on Networking*, 11.2 (2003): 259 – 272.

[3] A. Mokhtar and M. Azizoglu. "Adaptive techniques for routing and wavelength assignment in all-optical WANs." *IEEE 39th Midwest symposium on Circuits and Systems*, 3 (1996): 1195 – 1198.

[4] A. Mokhtar and M. Azizoglu. "Adaptive wavelength routing in all-optical networks." *IEEE/ACM Transactions on Networking*, 6.2 (1998): 197 – 206.

[5] E. Karasan and E. Ayanoglu. "Effects of wavelength routing and selection algorithms on wavelength conversion gain in WDM optical networks." *IEEE/ACM Transactions on Networking*, 6.2 (1998): 186 – 196.

[6] *GÉANT Home*. (Retrieved: July 15, 2005). <http://www.geant.net/>

[7] H. Zang, J. P. Jue, and B. Mukherjee. "A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks." *SPIE/Baltzer Optical Networks Magazine (ONM)*, 1.1 (2000): 47 – 60.

[8] J. Fang, W. He, and A. K. Somani. "Optimal light trail design in WDM optical networks." *IEEE ICC*, 3.20-14 (2004): 1699 – 1703.

[9] K. Sathyamurthy and S. Ramasubramanian. "Benefits of link protection at connection granularity." *Proceedings of IEEE International Conference on Broadband Networks*, October 2004, San Jose.

[10] L. Li and A. K. Somani. "Dynamic wavelength routing using congestion and neighborhood information." *IEEE/ACM Transactions on Networking*, 7.5 (1999): 779 – 786.

[11] M. T. Frederick and A. K. Somani. "A single-fault recovery strategy for optical networks using subgraph routing." *Proceedings of the 7th IFIP Working Conference on Optical Network Design and Modeling*, February 2003, Budapest.

[12] *Microsoft .NET*. (Retrieved: 3 April 2005). < http://www.microsoft.com/net/>

[13] P. Datta, M. T. Frederick, and A. K. Somani. "Sub-graph routing: A novel fault-tolerant architecture for shared-risk link group failures in WDM optical networks." *4th International Workshop on the Design of Reliable Communication Networks*, October 2003, Alberta.

[14] R. Gupta. *Protocols for Sharing Computing Resources and Dealing with Nodes' Selfishness in Peer to Peer Networks*. Dissertation. Iowa State University, 2005.

[15] R. Ramamurthy and B. Mukherjee. "Fixed-alternate routing and wavelength conversion in wavelength-routed optical networks." *IEEE/ACM Transactions on Networking*, 10.3 (2002): 351 – 367.

[16] R. Ramaswami and K. N. Sivarajan. *Optical networks: A practical perspective*. San Diego: Academic Press, 2002.

[17] R. Srinivasan and A. K. Somani. "Dynamic routing in WDM grooming networks." *Photonic Network Communications*, 5.2 (2003): 123 – 135.

[18] R. Srinivasan. "MICRON: A framework for connection establishment in optical networks." *Proceedings of OPTICOMM*, October 2003, Dallas.

[19] R. Srinivasan. *A generalized framework for analyzing time-space switched optical networks*. Dissertation. Iowa State University, 2002.

[20] S. Ramasubramanian and A. Harjani. "DIVERSION: A trade-off between link and path protection strategies." *Proceedings of 9th Conference on Optical Network Design and Modeling (ONDM)*, February 2005, Milan.

[21] S. Ramasubramanian and A. K. Somani. "Analysis of optical networks with heterogeneous grooming architectures." *IEEE/ACM Transactions on Networking*, 12.5 (2004): 931 – 943.

[22] S. Ramasubramanian and K. Sathyamurthy. "Supporting multiple protection strategies in optical networks." Submitted to *IEEE INFOCOM*, March 2005.

[23] S. Ramasubramanian. "On failure dependent protection in optical grooming networks." *Proceedings of International Conference on Dependable Systems and Networks*, June-July 2004, Florence.

[24] S. Ramesh, G. N. Rouskas, and H. G. Perros. "Computing blocking probabilities in multiclass wavelength routing networks." *ACM Transactions on Modeling and Computer Simulation*, 10.2 (2000): 87 – 103.

[25] T. H. Cormen, et. al. *Introduction to Algorithms*. 2nd ed. Cambridge: The Massachusetts Institute of Technology and McGraw-Hill, 2001.

[26] Tom Archer. *Inside C#*. Redmond: Microsoft Press, 2001.

APPENDIX

Table A- 1. Average blocking probability for NSFNet network.

Arrival rate	30	40	50	60	70	80
FDP	0.00075	0.00169	0.00359	0.00803	0.01907	0.04137
Backup multiplexing	0.00019	0.00298	0.01807	0.04802	0.09229	0.14774
None	0	0	0	0	0	0
L+1	0	0.00E+00	1.30E-04	0.00114	0.00543	0.016347

Table A- 2. Average blocking probability for 3x3 mesh torus network.

Arrival rate	60	70	80	90	100	110
Dedicated	0.0035	0.01136	0.02928	0.05618	0.08409	0.12323
FDP	0.17778	0.20384	0.22873	0.2503	0.26657	0.29134
Backup multiplexing	0.0036	0.01211	0.03019	0.05618	0.08386	0.12218
None	0	6.00E-05	0.00018	0.00028	0.00084	0.00208

Table A- 3. Average blocking probability in GEANTNet network.

Arrival rate	30	40	50	60	70	80
FDP	0.65762	0.65764	0.65769	0.65729	0.65891	0.65885
Backup multiplexing	0.65752	0.65751	0.6575	0.65699	0.65874	0.65924
None	3.00E-05	0.00013	0.00087	0.00265	0.00754	0.01546
L+1	0.74264	0.65668	0.6562	0.65666	0.65667	0.65579

Table A- 4. Normalized average effective utilization for NSFNet network.

Arrival rate	30	40	50	60	70	80
FDP	0.085048	0.113331	0.141275	0.168743	0.194251	0.215853
Backup multiplexng	0.083848	0.106289	0.123011	0.135714	0.146065	0.154536
None	0.085107	0.113518	0.141795	0.170254	0.198649	0.227117
L+1	0.085051	0.113391	0.141634	0.169307	0.195645	0.219193

Table A- 5. Normalized average effective utilization for 3x3 mesh torus.

Arrival Rate	60	70	80	90	100	110
Dedicated	0.15568	0.179868	0.201501	0.219527	0.236557	0.247974
FDP	0.128654	0.145073	0.160517	0.175227	0.190661	0.202767
Backup multiplexing	0.155676	0.17972	0.201269	0.219527	0.236576	0.248212
None	0.156284	0.18213	0.208118	0.233708	0.260241	0.285696

Table A- 6. Normalized average effective utilization for GEANTNet network.

Arrival Rate	30	40	50	60	70	80
FDP	0.030899	0.041025	0.051707	0.062087	0.072283	0.082486
Backup multiplexing	0.030908	0.041039	0.051732	0.062122	0.0723	0.082324
None	0.119438	0.158728	0.198796	0.237845	0.275772	0.311801
L+1	0.022004	0.041271	0.051751	0.062157	0.07198	0.082957